PHD

Minterm-interchange applications to digital circuit design.

Eris, E.

Award date:
1979

Awarding institution:
University of Bath

Link to publication

MINTERM-INTERCHANGE APPLICATIONS TO DIGITAL CIRCUIT DESIGN

submitted by E.Eris
for the degree of Ph.D.
of the University of Bath

1979

E. ERIS

ACKOWLEDGMENT

The author ackowledges a great debt to Dr. S.L.HURST, University of Bath,for his assistance,advice and encouragment throughout the period of research,and also Dr. C.R. EDWARDS,University of Bath,for profitable discussions and valuable advice.

This work was supported by Turkish Government between the period of March 1977 and February 1979,then by the University of Bath until the end of the work.

Last but not least I wish to thank to my family for their encouragment and support which made this research possible.

SUMMARY

The object of this thesis is to present firstly a new design technique for combinational logic circuits,and secondly a new state assignment method for the design of sequential machines.

The combinational design technique to be presented has been based upon two concepts.One of them is "minterm-interchange" by which it is possible to decompose a Boolean function into the exclusive-OR of two lower cost Boolean functions.The second concept concerns "simplest-threshold functions" which are logically simple.The cost of realisation of such functions should also be minimal in comparison with a wider class of functions.The techniques based upon these concepts are implemented in the spectrum domain,which is briefly surveyed in Chapter 1.Chapter 2 details the mathematics of the minterm-interchange operation in the spectrum domain.Chapter 3 is concerned with the design algorithm and also with the conditions of applicability of this technique to the Boolean function realisation.Also in Chapter 3,the combination of the already known spectral translation technique and the new minterm-interchange technique has been considered.

Finally in Chapter 4 two basic concepts used in combinational logic design are applied in the proposed new state assignment technique for sequential machine design.This new state assignment is implemented by means of a binary tree.

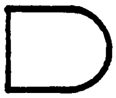
NOTATIONS AND DEFINITIONS

n	Number of independent input variables.
ω	Index of Walsh function, $\omega = \sum_{s=o}^{n-1} \omega_s 2^{n-s-1}$.
$x(x_1,x_2,..x_n)$	Argument $x = \sum_{s=1}^{n} x_s 2^{n-s}$.
$m_i(m_i^1,....m_i^n)$	Minterm identification number same as x.
$W_\omega(x)$	Walsh function.
$R_s(x)$	Rademacher function , $1 \leqslant s \leqslant n$.
$\Vert\omega\Vert$	Index weight i.e. number of '1's in the binary expansion of ω .
Mod 2 operation	Exclusive-OR operation for binary case.
$\underset{\sim}{f}(x_1,x_2,...x_n)=\underset{\sim}{f}(x)$	$\langle 0,1 \rangle$ Boolean function vector of 2^n entries in decimal order (corresponds to truth table of the Boolean function.
$\underset{\sim}{F}(x)$	$\langle +1,-1 \rangle$ Boolean function vector in decimal order.
I	Unit matrix.
$\underset{\sim}{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$	Unit column matrix.
T_{R-W}	Rademacher-Walsh ordered transform matrix.
T_H	Hadamard ordered transform matrix.
T	with no subscript, Rademacher-Walsh unless otherwise defined.

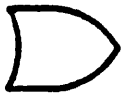
$\underset{\sim}{S}(x) = \underset{\sim}{S}$	Spectrum of a Boolean function,	
R_i	Spectral coeffecients in the spectrum $\underset{\sim}{S}$. $i= 0,1,..n,12,13,..1n,..,1\ \ n$.	
w_i'	Weights of the inputs of threshold gates.	
( ]	Left end open interval.	
[ )	Right end open interval.	
$\underset{\sim}{A} * \underset{\sim}{B}$	Element by element product of the two vectors $\underset{\sim}{A}$ and $\underset{\sim}{B}$.	
$\wedge$	Logic AND	These notations are used only in Chapter 2,to distinguish them from mathematical (.) and (+).Rest of the chapters,(.) and (+) notations are used for logic AND and OR,as usual.
$\vee$	Logic OR	
$\underset{\sim}{\delta}$	Interchanged function.	see detailed definition in Chapter 2.2.
$\underset{\sim}{\alpha}$	True-function.	
$\underset{\sim}{\beta}$	False-function.	
$\underset{\sim}{\gamma}$	Changer function.	

Throughout the thesis only the true logic "1" minterms are shown in the Karnaugh map of all Boolean functions.

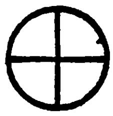
SYMBOLS



AND gate.



OR gate.



Exclusive-OR gate.

CONTENTS

CHAPTER 1. INTRODUCTION TO SPECTRAL TRANSFORMATION AND ITS APPLICATIONS

1.1. Orthogonal Transformations

1.1.1. Orthogonal Transformation as a Finite Series and its Properties

1.1.2. Orthogonal Transformation in Matrix Form and its Properties

1.2. The Spectrum of a Boolean Function

1.2.1. Evaluating the Spectrum of a Boolean Function

1.2.2. Some Operations in the Spectrum Domain

1.3. The Relationship Between Spectral Coeffecients and Minterm Distribution

1.4. Application of Spectral Translation to Combinational Logic Design by Threshold and Embedded Threshold Specifications

CHAPTER 1 INTRODUCTION TO SPECTRAL TRANSFORMATION AND ITS APPLICATIONS

In this thesis we will refer often to the spectral domain[1,2,3,4] especially in Chapters two and three.This chapter is therefore devoted to the introduction of spectral transformations with its particular application to combinational circuit design and the special class of threshold functions called " embedded ".

In 1922 Rademacher published a set of orthogonal functions taking the value ∓ 1 in the interval (0,1).This set of orthogonal functions however was incomplete,that is a finite set of such functionsdoes not form a subgroup.Working independently,in 1923, Walsh [6] published a set of orthogonal functions which,in addition to forming a complete set, have Rademacher functions as a generating set.That is to say,any complete set of Walsh functions may be generated from a suitable set of Rademacher functions[7,8] as will be shown in section 1.1.Because the Walsh functions have properties analogous to trigonemetric functions,considerable reaserch has gone into employing "Walsh waves "for the transmission of sampled-data digital information.Other areas of applications have been in the field of signal filtering and pattern recognition.Both of this areas are entirely outside our particular interest in this thesis,and will not again be mentioned.

In the field of logic design the Walsh functions appear to have been employed relatively little.Chow [9] showed however that certain numerical parameters were suffecient to characterise all linearly-seperable functions,that is threshold fuctions.These threshold functions and their applications were further developed by Lewis and Coates [10], Sheng [11] and Murago [12]. Dertouzos [1] showed that the Chow parameters were infact a subset of the Walsh transform coeffecients. Dertouzos also developed operators for the manipulation of these coeffecients to facilitate threshold logic synthesis. Ito [13] considered the application of Walsh functions to the recognition of binary valued functions on statistical basis. Hurst [14] has considered the general possibities of the application of Walsh functions to the synthesis of binary functions both in terms of threshold and conventional logic circuitry. Finally Edwards [15] applied spectral techniques to combinational circuit design.

1.1 Orthogonal Transformations

1.1.1 Orthogonal Transformation as a Finite Series and Their Properties

The Walsh functions are step functions on the interval $[0,2^n)$

$$W_\omega(x) = (-1)^{\sum_{s=1}^{n} \omega_{s-1} x_s} \qquad \ldots 1.1$$

where ω_s and x_s are determined by the binary expansions of ω and x

$$\omega = \sum_{s=0}^{n-1} \omega_s \, 2^{n-s-1} \qquad 0 \leqslant \omega \leqslant 2^n-1 \qquad \ldots 1.2$$

$$x = \sum_{s=1}^{n} x_s \, 2^{n-s} \qquad 0 \leqslant x \leqslant 2^n-1 \qquad \ldots 1.3$$

where ω is called the index, and the number of ' 1 ' s in the binary expantion of ω (i.e. $\|\omega\| = \sum_{s=1}^{n} \omega_{n-s}$) will be called the weight of the index. x is called the argument or the minterm identification number(sometimes shown as m_i).

The Walsh functions defined by equation(1.1,2,3) are discrete functions defined at the points $0,1,\ldots,2^n-1$ of the interval $[0,2^n)$ and are illustrated in figure 1.1 for n=3.

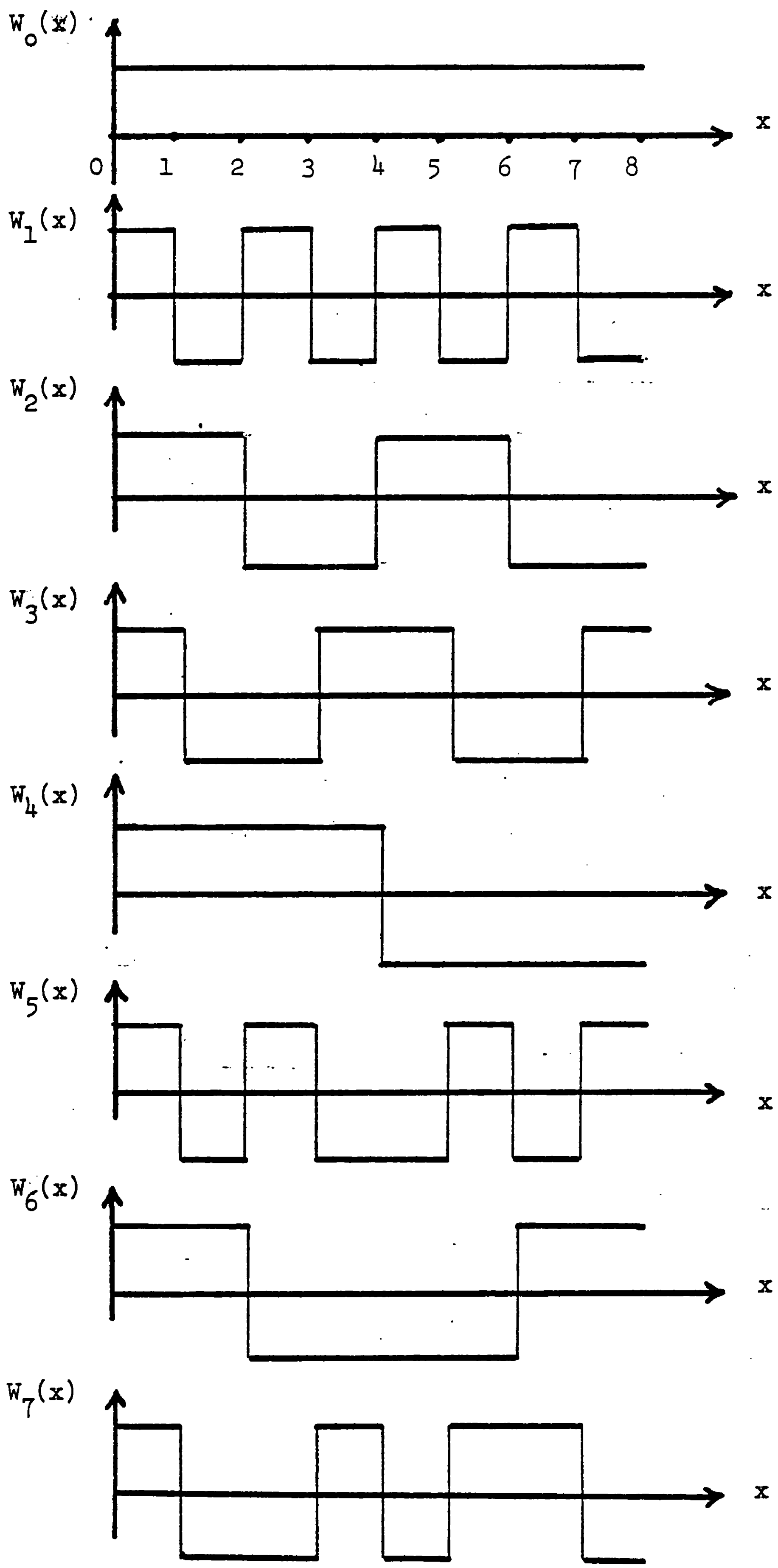


Figure 1.1 The Walsh Functions for n=3 variables.

Consider the subset of Walsh functions with the indices of 2^i $(i=0,1,\ldots,n-1)$

$$R_s(x)= W_{2^{s-1}}(x) = (-1)^{x_s(x)} \qquad 1 \leqslant s \leqslant n \qquad \ldots 1.4$$

These functions $R_s(x)$ are known as the Rademacher functions from which the rest of Walsh functions can be determined as follows:

$$W_\omega (x) = \prod_{s=1}^{n} \left[R_s(x)\right]^{\omega_{s-1}} \qquad \ldots 1.5$$

That is Rademacher functions form one basis for generating the Walsh functions.

Some of the properties of Walsh functions are stated below without formal proof (for proofs see reference 1.3)

Theorem 1.1 The Walsh functions form a complete orthogonal system.That is

$$\sum_{x=0}^{2^n-1} W_t(x) \; W_\omega(x) = \begin{cases} 2^n & \text{if } t=\omega \\ 0 & \text{if } t\neq\omega \end{cases} \qquad \ldots 1.6$$

Theorem 1.2 For any ω, $0 \leqslant \omega \leqslant 2^n-1$

$$\sum_{x=0}^{2^n-1} W_\omega (x) = \begin{cases} 2^n & \text{if } \omega =0 \\ 0 & \text{if } \omega \neq 0 \end{cases} \qquad \ldots 1.7$$

viz. the sum of the entries of these functions equals '0' except the function with index ω =0.

Theorem 1.3 The Walsh functions are symmetric for index and argument

$$W_\omega (x) = W_x(\omega) \qquad \ldots 1.8$$

viz. interchanging index and argument gives the same value.

Theorem 1.4 For any ω , τ , $x \in \{0,1,\ldots,2^n-1\}$

$$W_\omega(x \oplus \tau) = W_\omega(x) \; W_\omega(\tau) \quad \text{Mod } 2. \qquad \ldots 1.9$$

viz. the multiplication of any two entries of a Walsh function gives another entry in the same function. For example x=5, τ =3, n=4, ω =6

$$W_6(5 \oplus 3) = W_6(5)\, W_6(3)$$

$$W_6(6) = W_6(5)\, W_6(3)$$

Theorem 1.5 The group of Walsh functions is isomorfic to the group of linear Boolean functions* (see reference 3).

For example the Walsh function $W_5(x)$, n=3

$$W_5(x) = [1 \;\; -1 \;\; 1 \;\; -1 \;\; -1 \;\; 1 \;\; -1 \;\; 1]^t$$

corresponds to the linear Boolean function

$$\underset{\sim}{x}_1(x) \oplus \underset{\sim}{x}_2(x) = [0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 0]^t.$$

Translation between the Walsh functions and related linear Boolean functions can be achieved simply by substituting '1' in the Walsh function for '0' in the Boolean function and similiarly ' -1' for '1'.Mathematically

$$\underset{\sim}{W}_\omega(x) = -2\, \underset{\sim}{f}(x) + \underset{\sim}{1} \qquad \ldots 1.11$$

1.1.2 Orthogonal Transformation in Matrix Form and its Properties

If the above defined Walsh functions are chosen as the rows of a matrix, we obtain a ($2^n \times 2^n$) order orthogonal matrix.A particular row ordering of the Walsh functions defines a particular variant of such matrices (for further details see reference 16).We shall consider only the two row orderings (a) and (b) following.

---

* Linear Boolean function: A Boolean function f(x) is said to be linear if it may be expressed as

$$\underset{\sim}{f}(x) = c_0 \underset{\sim}{x}_1 \oplus c_1 \underset{\sim}{x}_2 \oplus \ldots \oplus c_{s-1} \underset{\sim}{x}_s \oplus \ldots \oplus c_{n-1} \underset{\sim}{x}_n \qquad \ldots 1.10$$

where c_s, $x_s \in 0,1$

a) The Rademacher-Walsh ordered transform matrix T_{R-W}:

This matrix can be determined by ordering the Walsh functions from the low index weight towards the high index weight. In Figure 1.2 the Rademacher-Walsh ordered transform matrices are shown for n=2,3.

Correspond. linear Boolean fun. | Weights of indices

$$T_{R-W}= \begin{array}{c} W_0(x)\\ W_2(x)\\ W_1(x)\\ W_3(x)\end{array} \begin{bmatrix} 1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1\\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{array}{l} \dots \\ \dots x_1\\ \dots x_2\\ \dots x_1\oplus x_2 \end{array} \quad \begin{array}{c} \|\omega\|=0\\ \|\omega\|=1\\ \\ \|\omega\|=2 \end{array}$$

$$T_{R-W}= \begin{array}{c} W_0(x)\\ W_4(x)\\ W_2(x)\\ W_6(x)\\ W_1(x)\\ W_5(x)\\ W_3(x)\\ W_7(x)\end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{array}{l} ..\\ ..x_1\\ ..x_2\\ ..x_3\\ ..x_1\oplus x_2\\ ..x_1\oplus x_3\\ ..x_2\oplus x_3\\ ..x_1\oplus x_2\oplus x_3 \end{array} \quad \begin{array}{c} \|\omega\|=0\\ \\ \|\omega\|=1\\ \\ \\ \|\omega\|=2\\ \\ \|\omega\|=3 \end{array}$$

Figure 1.2 Rademacher-Walsh transform matrices for n=2 and n=3.

some of the properties of the Rademacher-Walsh ordered transform matrix are:

i)The scalar product of any two rows of this matrix is equal to '0'.

ii)The sum of the elements of any row(or column),except first, is equal to '0' .

iii)It is orthogonal,i.e $T^{-1} = \frac{1}{2^n} T^t$.

iv)The element-by-element product of any two rows for which $\|\omega\|=1$ gives another row in $\|\omega\|=2$ and any three rows for which

$\|\omega\|$ =1 gives another row in $\|\omega\|$ =3 and so on.

b)Secondly, the Hadamard ordered transform matrix T_H:

It is determined by ordering the Walsh functions in decimal order.The n th ordered Hadamard matrix can be obtained from the(n-1) th ordered Hadamard matrix by the formula below(also called a Kronecker δ expansion).

$$T_{n\times n} = \begin{bmatrix} T_{(n-1)\times(n-1)} & T_{(n-1)\times(n-1)} \\ T_{(n-1)\times(n-1)} & -T_{(n-1)\times(n-1)} \end{bmatrix} \qquad ..1.12$$

Hadamard ordered transform matrices for n=2,3,4 are illustrated in Figure 1.3.

$$T_{H=2} = \begin{array}{c} W_0(x) \\ W_1(x) \end{array} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{array}{l} \\ ...x_1 \end{array}$$

$$T_{H=3} = \begin{array}{c} W_0(x) \\ W_1(x) \\ W_2(x) \\ W_3(x) \end{array} \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right] \begin{array}{l} \\ ..x_2 \\ ..x_1 \\ ..x_1 \oplus x_2 \end{array} = \begin{bmatrix} T_{H=2} & T_{H=2} \\ T_{H=2} & -T_{H=2} \end{bmatrix}$$

$$T_{H=4} = \begin{array}{c} W_0(x) \\ W_1(x) \\ W_2(x) \\ W_3(x) \\ W_4(x) \\ W_5(x) \\ W_6(x) \\ W_7(x) \end{array} \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right] \begin{array}{l} \\ ..x_3 \\ ..x_2 \\ ..x_2 \oplus x_3 \\ ..x_1 \\ ..x_1 \oplus x_3 \\ ..x_1 \oplus x_2 \\ ..x_1 \oplus x_2 \oplus x_3 \end{array} = \begin{bmatrix} T_{H=3} & T_{H=3} \\ T_{H=3} & -T_{H=3} \end{bmatrix}$$

Figure 1.3 Hadamard ordered transform matrices for n=2,3,4.

Some of the properties of Hadamard ordered transform matrix T_H are;

i) The scalar product of any two rows (or columns) is equal to '0'.

ii) The sum of elements of any row (or column), except the first, is equal to '0'.

iii) It is orthogonal i.e. $T^{-1} = \frac{1}{2^n} T^t$.

iv) It is symmetric.

v) The element-by-element product of any rows (or columns) gives another row (column) in the matrix.

For details of other orthogonal matrices such as Haar see references 3,4,16,17,18,19.

1.2 The Spectrum of a Boolean Function

1.2.1 Evaluating the Spectrum of a Boolean Function

In this thesis we shall use the Rademacher-Walsh ordered transform matrix. Henceforth T_{R-W} will be denoted by T . Let us now use this matrix to transform the binary Boolean function to the spectrum domain. Suppose we have the three variable Boolean function

$$f(x)=f(x_1,x_2,x_3) = \bar{x}_2 + \bar{x}_1 \bar{x}_3 \ .$$

Its truth table is shown in Figure 1.4 together with conversion of the output from the f(x) < 0,1 > values to the F(x) < 1, -1 > values. This is the same numerical conversion as previously noted in theorem 1.5 with the equation 1.11.

Minterms	x	x_1	x_2	x_3	f(x)	F(x)
m_0	0	0	0	0	1	-1
m_1	1	0	0	1	1	-1
m_2	2	0	1	0	1	-1
m_3	3	0	1	1	0	1
m_4	4	1	0	0	1	-1
m_5	5	1	0	1	1	-1
m_6	6	1	1	0	0	1
m_7	7	1	1	1	0	1

Figure 1.4 Example function $f(x)=\bar{x}_2+\bar{x}_1 \bar{x}_3$ and its conversion from < 0,1> values to < 1,-1> values.

The spectrum of f(x), $\underset{\sim}{S}$ is derived as follows

$$T\,\underset{\sim}{F}(x) = \underset{\sim}{S} \qquad \ldots 1.13$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} R_o \\ R_1 \\ R_2 \\ R_3 \\ R_{12} \\ R_{13} \\ R_{23} \\ R_{123} \end{bmatrix}$$

The spectrum of the given function f(x) has the spectral coeffecients:

-2 -2 -6 -2 2 -2 2 2

(written in descending order of the vector $\underset{\sim}{S}$}.These coeffecients are labelled R_o, R_1, ... etc. thus giving for our example the detailed resultant coeffecients values:

$$\underbrace{\begin{matrix} R_o \\ -2 \end{matrix}}_{\text{zero order spectral coef.}} \quad \underbrace{\begin{matrix} R_1 & R_2 & R_3 \\ -2 & -6 & -2 \end{matrix}}_{\text{first order spectral coeffecients}} \quad \underbrace{\begin{matrix} R_{12} & R_{13} & R_{23} \\ 2 & -2 & 2 \end{matrix}}_{\text{Second order spectral coeffecients}} \quad \underbrace{\begin{matrix} R_{123}. \\ 2 \end{matrix}}_{\text{Third order spectral coeffecients}}$$

The spectral coeffecients corresponding to the $\|\omega\| = 1$ index weight are called "first-order spectral coeffecients", and those corresponding to the $\|\omega\| = 2$ "second-order spectral coeffecients" and pro.rata.

We may interpret the resulting values of the spectral coeffecients in $\underset{\sim}{S}$ as follows:

$$R_0 = \{(\text{ number of '+1' in } F(x)) - (\text{ number of '-1' in } F(x)\}$$
$$\equiv \{(\text{number of '0' in } f(x) - (\text{ number of '1' in } f(x))\} \ldots 1.14$$

$R_i(1 \leq i \leq n) = \{$(number of agreements between F(x) and the corresponding Rademacher function $R_i(x)$) - (number of disagreements between F(x) and the corrosponding Rademacher function $R_i(x)$)$\}$

$\equiv \{($ number of agreements between $f(x)$ and the input variable x_i) - (number of disagreements between $f(x)$ and x_i)$\}$ ...1.15

R_{ij} ($1 \leq i < j \leq n$)= $\{($ number of agreements between $F(x)$and $\underset{\sim}{R}_i(x)*\underset{\sim}{R}_j(x)$- (number of disagreements between $F(x)$ and $\underset{\sim}{R}_i(x)*\underset{\sim}{R}_j(x)\}$

$\equiv \{$(number of agreements between $f(x)$ and the function $x_i \oplus x_j$) - (number of disagreements between $f(x)$ and $x_i \oplus x_j)\}$. ...1.16

Similiarly for higher order spectral coeffecients.

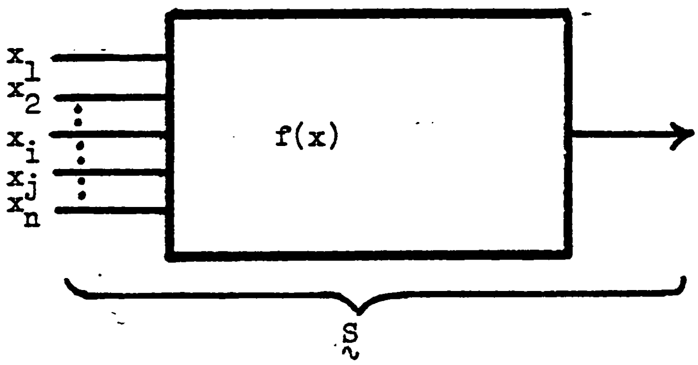
This interpretation of spectral coeffecients will be used in section 1.3 when we consider the relationship between spectral coeffecients and minterm distributions.

1.2.2 Some Operations in Spectral Domain

We shall consider some basic Boolean operations and their corresponding operations in spectrum domain.The circuit implementations of all these operations will be stated without detail.For further reading see references 1,4,15,21.

Operation 1 Interchange of variables x_i and x_j,$i \neq j$:

The new spectrum $\underset{\sim}{S}'$ may be generated from the original spectrum $\underset{\sim}{S}$ under the interchange of x_i and x_j if in $\underset{\sim}{S}$ the subscript"i" is replaced by subscript "j" and vice-versa.Figure 1.5 shows the circuit implementation of this operation.



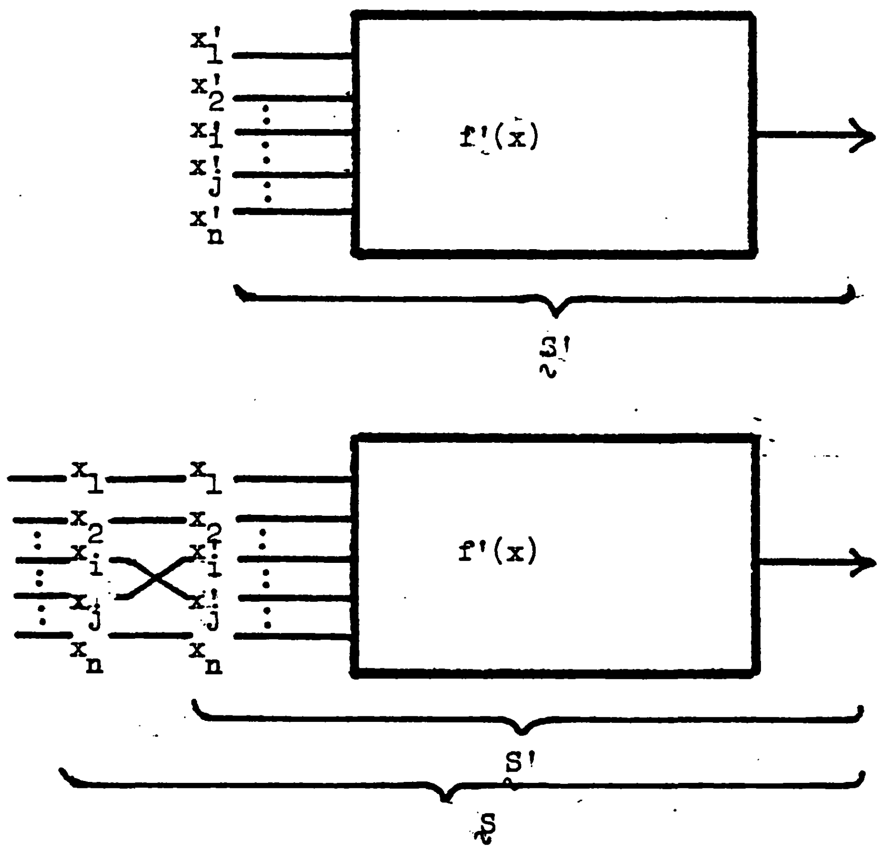


Figure 1.5 The circuit implementation of independent variable interchange.

Operation 2 Complementation of the variable x_i:

The new spectrum $\underset{\sim}{S}'$ under the complementation of variable x_i if in $\underset{\sim}{S}$ the spectral coeffecients having subscripts containing"i" are changed in sign. Figure 1.6 shows the circuit implementation of this operation.

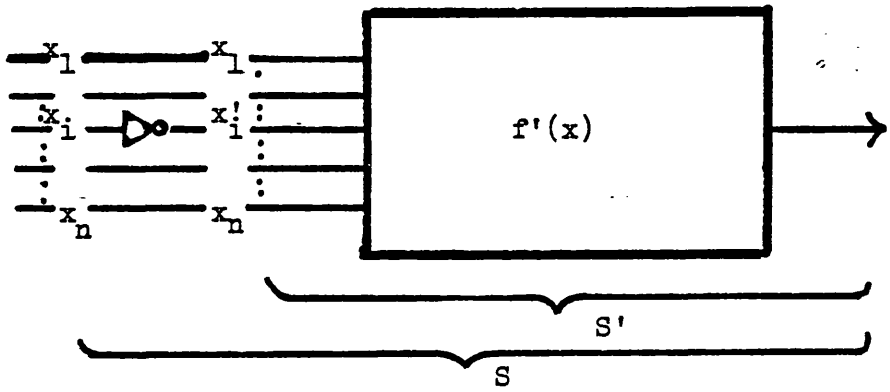


Figure 1.6 The circuit implementation of the complementation of the variable x_i.

<u>Operation 3</u> The generation of the dual of a Boolean function:

Given function $f(x_1,x_2,..x_i..x_n)$ having a spectrum $\underset{\sim}{S}$ generate the dual function $\overline{f(\bar{x}_1,\bar{x}_2,..\bar{x}_i..\bar{x}_n)}$ having a spectrum S'. The new spectrum $\underset{\sim}{S}'$ may be generated from the original spectrum $\underset{\sim}{S}$ under the above operation if in $\underset{\sim}{S}$ the even-ordered spectral coeffecients are changed in sign.Note R_o is even-ordered.Figure 1.7 shows the circuit implementation of this operation.

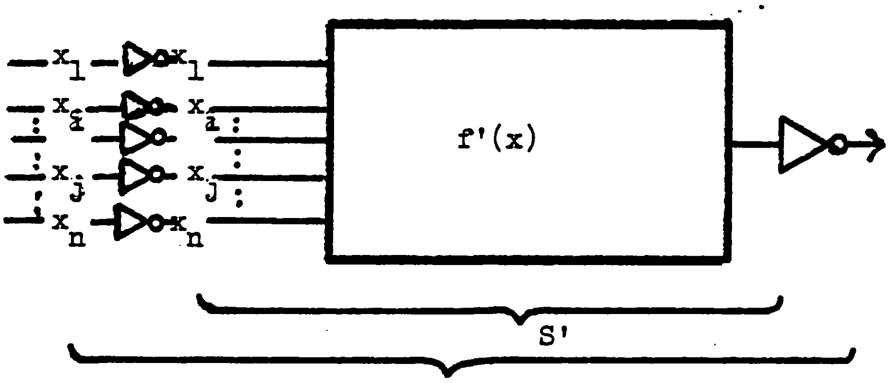


Figure 1.7 The circuit implementation of the dual of the Boolean function $f(x)$.

<u>Operation 4</u> The generation of the complement of a Boolean function:

Given the function $f(x_1,x_2,..x_i..x_n)$ having the spectrum $\underset{\sim}{S}$ a function $\overline{f(x_1,x_2,..x_i..x_n)}$ having the spectrum $\underset{\sim}{S}'$.The new spectrum $\underset{\sim}{S}'$ may be generated from the original spectrum $\underset{\sim}{S}$ under the complementation of the function if all spectral coeffecients in $\underset{\sim}{S}$ are changed in sign. Figure 1.8 shows the circuit implementation of this operation.

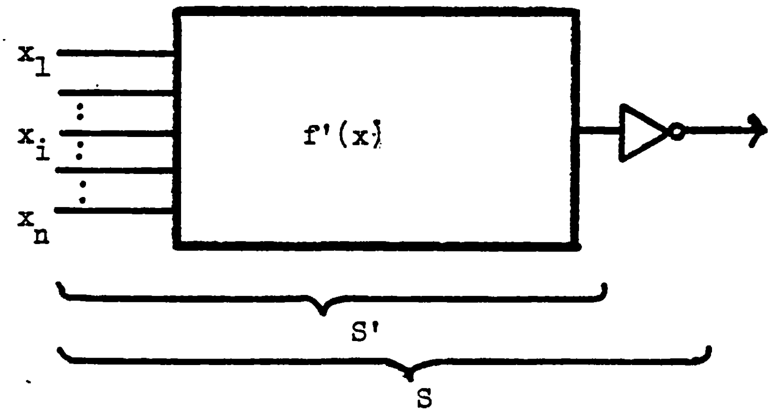
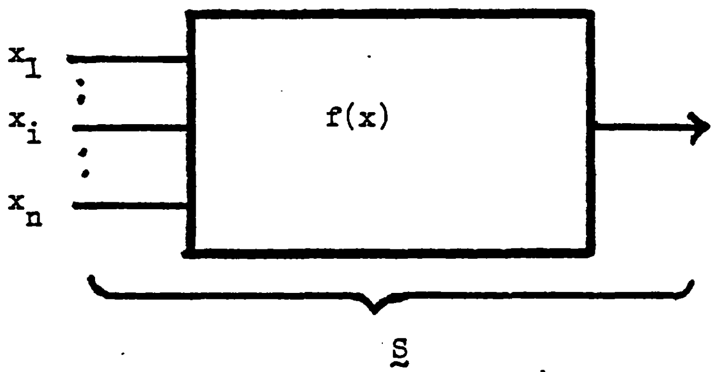


Figure1.8 The circuit implementation of the complementation of the Boolean function $f(x)$.

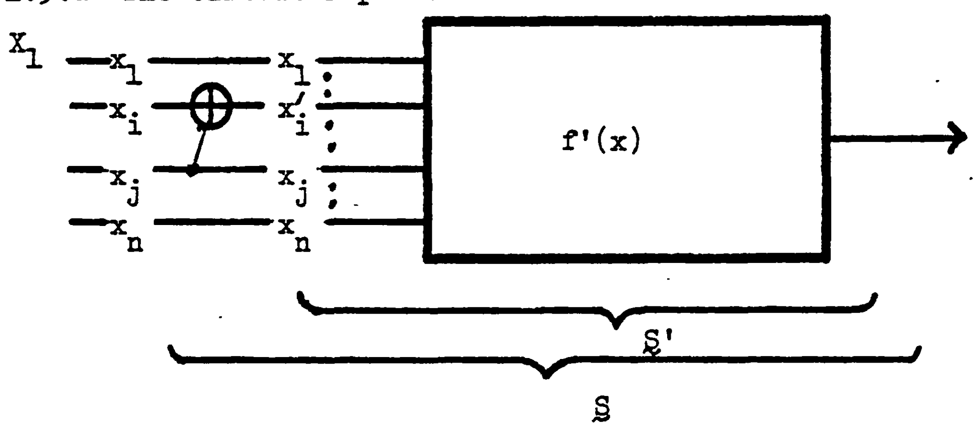
Operation 5: Spectral translation:

If in a Boolean function $f(x_1,..x_i..x_n)$ having a spectrum $\underset{\sim}{S}$, x_i is replaced by$(x_a \oplus x_b \oplus .. \oplus x_h) \oplus x_i$,where the set of subscripts $\{a,b,...h\}$ is denoted by K,then the spectrum $\underset{\sim}{S}'$ of the new function is generated from the spectrum $\underset{\sim}{S}$ if in every subscript of the spectral coeffecients of $\underset{\sim}{S}$ containing 'i',the members of K are deleted if they exist and appended if they do not.

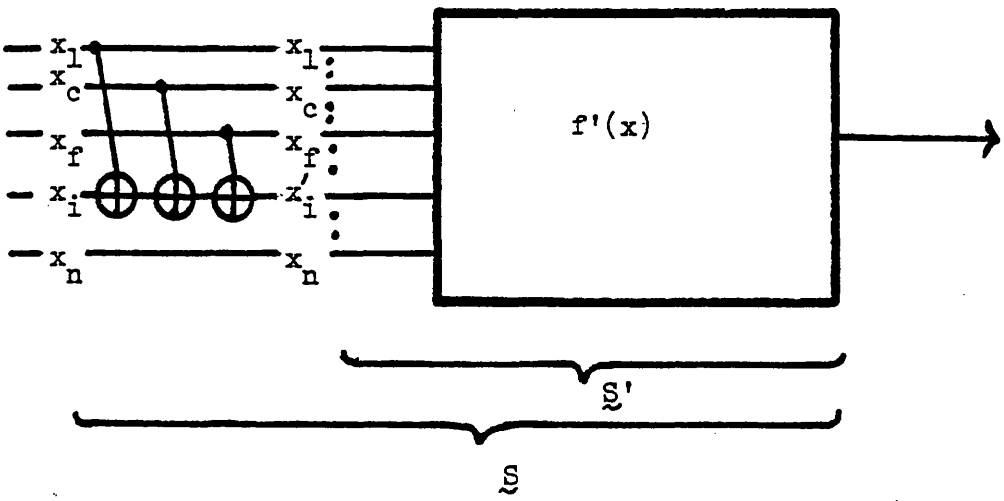
Figure 1.9 a shows the implementation of the Boolean function f(x) in terms of logic circuitry.Suppose that it is required to replace x_i by $x_i' = x_i \oplus x_j$ .This is accomplished by means of an exclusive-OR gate and produces a new logic module $f'(x_1,..x_i'..x_n)$ as shown in Figure 1.9.b. Figure 1.9.c shows the implementation of this operation for the variable x_i replaced by $x_i' = (x_1 \oplus x_c \oplus x_f) \oplus x_i$.



1.9.a The circuit implementation of the Boolean function f(x).



1.9.b The circuit implementation of the operation $x_i' = x_i \oplus x_j$ of the Boolean function $f(x_1,..x_i..x_n)$.



1.9.c The circuit implementation of the operation $x_i' = x_i \oplus (x_1 \oplus x_c \oplus x_f)$ of the Boolean function $f(x_1 .. x_i .. x_n)$.

Figure 1.9 The circuit implementations of some spectral translations.

1.3 The Relationship Between Spectral Coeffecients and Minterm Distributions

It follows from the equations 1.14,15,16 that

$$R_{ij..n} = n_a - n_d \qquad ...1.17$$

where n_a is the number of agreements between the defining function and the function $x_i \oplus x_j \oplus .. \oplus x_n$, and n_d is the number of disagreements between the defining function and $x_i \oplus x_j \oplus .. \oplus x_n$.In addition,the equation below is always true

$$n_a + n_d = 2^n \qquad ...1.18$$

Since the defining function must either agree or disagree with $x_i \oplus x_j ... x_n$ at all n-tuples, substituting for n_d in equation 1.17we will have

$$R_{ij..n} = n_a - (2^n - n_a),$$
$$= 2n_a - 2^n ,$$

whence

$$n_a = \frac{R_{ij..n} + 2^n}{2} . \qquad ...1.19$$

In similar manner:

$$n_d = \frac{2^n - R_{ij..n}}{2} . \qquad ...1.20$$

for all spectral coeffecients with the exception of the zero ordered coeffecient.

Hence

$$n_a = Z+V \text{ ,} \qquad \ldots 1.21$$

where Z is the number of true minterms of the defining function in the space* $x_i \oplus x_j \oplus \ldots \oplus x_n = 1$ and V is the number of false minterms of the defining function in the space $x_i \oplus x_j \oplus \ldots \oplus x_n = 0$. Since the space covered by $x_i \oplus x_j \oplus \ldots \oplus x_n = 0$ is $(2^n/2)$ n-tuples, it follows that the number of true minterms in this space is $(2^n/2)-V$, and thus the total number of true minterms of the defining function, u is given by

$$u = Z + \left(\frac{2^n}{2} - V\right). \qquad \ldots 1.22$$

Substituting for V in equation 1.22 from equation 1.21

$$u = Z + (2^n/2) - n_a + Z \text{ ,}$$

$$= 2Z + (2^n/2) - n_a \text{ .} \qquad \ldots 1.23$$

Substituting for n_a in equation 1.23 from equation 1.19

$$u = 2Z + (2^n/2) - \frac{R_{ij..n} + 2^n}{2} \text{ ,}$$

$$= 2Z - \frac{R_{ij..n}}{2} \text{ ,}$$

whence

$$Z = \frac{1}{4}\,(2u + R_{ij..n}) \text{ .} \qquad \ldots 1.24$$

Equation 1.24 shows the direct relationship between the value of the spectral coeffecient $R_{ij..n}$ and the true minterm distribution Z.

1.4 Application of Spectral Translation to Combinational Logic by Threshold and Embedded-Threshold Specifications

Dertouzos [1] has shown that a threshold function is uniqule characterised by the values of the first (n + 1) spectral coeffecients. These infact are Chow parameters[9]. Morever these coeffecients may appear in any order with any sign. All threshold functions are linearly-seperable and, because the detection of linearly-seperable functions is a complex procedure, look-up tables of such functions have been

---

* see Appendix D for these spaces when n=4.

prepared[1,22] . In these tables the first (n +1) spectral coeffecients of each threshold function appear in ascending order of magnitute and are positive.These vectors are sufficient to characterise all n th order threshold functions,and are called the positive-canonic -characteristic vectors(or sometimes simply "canonic vectors").In order to establish if a given function is a threshold function it suffices to derive and arrange the first (n + 1) spectral coeffecients of the given function in ascending order of magnitute, change all negative coeffecients to positive,and determine if this characteristic vector appears in the tables of positive-canonic-characteristic vectors.

In order that the threshold gate corresponding to a particular canonic vector may be found, it is necessary to determine the weights associated with that vector.These threshold weights are also normally tabulated in the canonic vector look-up tables.A representetive set of such tables appears Appendix B.

The use of such tables is best illustrated by means of an example.Consider the fourth-order function of figure 1.10.The first n+1 spectral coeffecients of this function are:

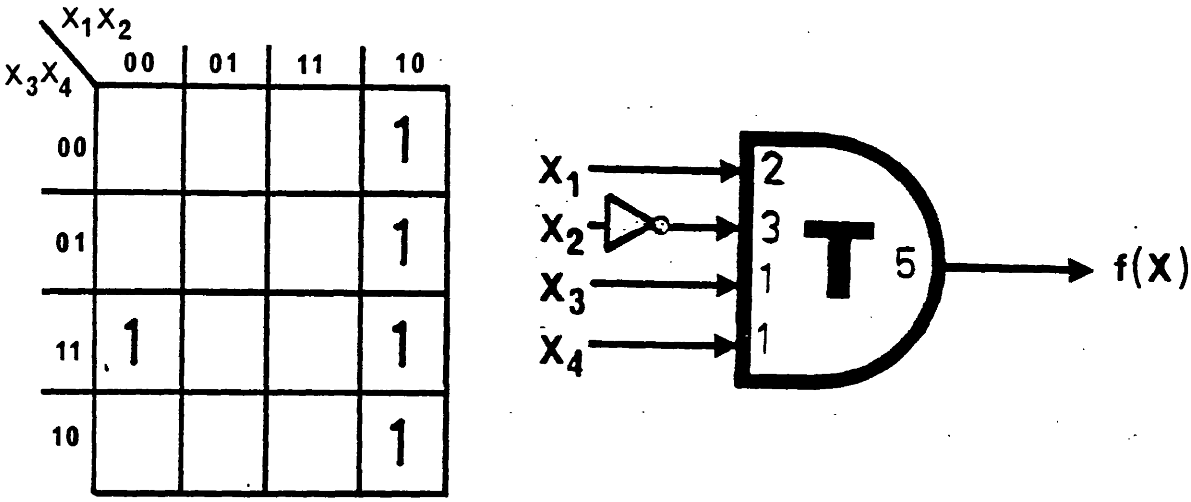
R_0	R_1	R_2	R_3	R_4
6	6	-10	2	2

Rearranging these coeffecients into ascending order of magnitude and changing all negative signs to positive ,the vector

10 6 6 2 2

is obtained. Inspection of Appendix B for n=4 shows that this characteristic vector indeed is present,and defines a threshold function for which :

Canonic-vector	10	6	6	2	2
Weights W:	3	2	2	1	1 .



$f = x_1\bar{x}_2 + \bar{x}_2x_3x_4$

Figure 1.10 An example of using threshold function table.

Now because there is a one-to-one correspondence between each weight and associated number of the characteristic vector,both in magnitude and sign,it is possible to re-express the original function in terms of the weights by re-arrangment and change of sign as appropriate.In this example

R_0	R_1	R_2	R_3	R_4	
6	6	-10	2	2	are the original coeffecients and
w'_0	w'_1	w'_2	w'_3	w'_4	
2	2	-3	1	1	are the corresponding weights.

From the weights,the parameters of the required threshold gate may now be calculated as fallows:

The input weighting for each gate-input is given by:

Weighting at input x_i is equal to w'_i $\quad 1 \leqslant i \leqslant n$, ...1.25

the output weightings of the gate is given by:

$$\text{Weighting at output} = \frac{1}{2}\left\{ \left(\sum_{i=1}^{n} |w'_i| \right) + w'_0 + 1 \right\} \qquad \ldots 1.26$$

As threshold gates with a negative weight capability will not be considered,it is important to note that if some w'_i are negative the respective input may be complemented and corresponding weight changed in sign.In this particular example therefore w_2' is changed in sign and an inverter is placed before input x_2.

From equation 1.26,the weighting at the output of this gate is $\frac{1}{2}$ (7 + 2 +1) = 5.The required gate is shown in figure 1.10.The description of the operation of this gate is now straight-forward. Clearly if x_1=1 and $\bar{x}_2$=1 then the output threshold of 5 will be equalled since x_1 is weighted 2 and $\bar{x}_2$ is weighted 3.The gate will thus give an output of '1'.Similarly the gate will also give an output of '1' if $\bar{x}_2=1,x_3=1,x_4=1$ the sum of the weights at the input again being 5 which is equal to output threshold 5.The result can be checked from the Karnaugh map of the function shown in Figure 1.10.

For more detailed treatment see references 1,2,3.

The role of spectral translation in the synthesis of Boolean functions,by means of threshold functions,is now considered by means of simple example .

Consider the function shown by Karnaugh map of Figure 1.11a. The spectrum of this function is as follows:

R_0	R_1	R_2	R_3	R_4	R_{12}	R_{13}	R_{14}
6	-2	2	-6	-2	-6	2	-2
R_{23}	R_{24}	R_{34}	R_{123}	R_{124}	R_{134}	R_{234}	R_{1234}
6	-6	2	-2	-6	2	-2	-2

If the first (n + 1) spectral coeffecients of this function are ordered by magnitude and rendered positive,the result is:

6 6 2 2 2

which does not appear in the look-up tables of positive-characteristic vectors,viz the function is not threshold function.Let us apply spectral translation to generate a new spectrum $\underset{\sim}{S}'$ from the above spectrum $\underset{\sim}{S}$,using $R_1' \Leftrightarrow R_{12}$(operation 5 section 1.2.2)

$X_3X_4 \backslash X_1X_2$	00	01	11	10
00	1		1	
01		1	1	
11				
10	1			

a)Original function

$X_3X_4 \backslash X_1X_2$	00	01	11	10
00	1	1		
01		1	1	
11				
10	1			

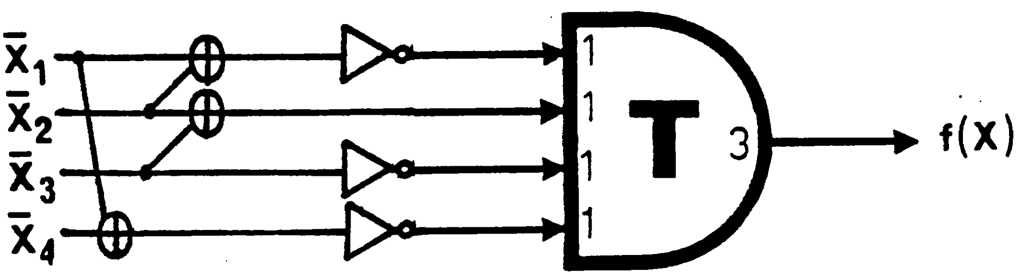
b) Operation $x_1 \Leftrightarrow x_1 \oplus x_2$ on the original function

$X_3X_4 \backslash X_1X_2$	00	01	11	10
00	1	1		
01		1	1	
11				
10		1		

c) Operation $x_2 \Leftrightarrow x_2 \oplus x_3$ on the function in b.

$X_3X_4 \backslash X_1X_2$	00	01	11	10
00	1	1	1	
01		1		
11				
10		1		

d) Operation $x_4 \Leftrightarrow x_1 \oplus x_4$ on the function in c.



e)Threshold realisation of the example function f.

Figure 1.11 Example for spectral translation design technique.

R_0'	R_1'	R_2'	R_3'	R_4'	R_{12}'	R_{13}'	R_{14}'
6	-6	2	-6	-2	-2	-2	-6
R_{23}'	R_{24}'	R_{34}'	R_{123}'	R_{124}'	R_{134}'	R_{234}'	R_{1234}'
6	-6	2	2	-2	-2	-2	2

see Figure 1.11.b.

Using this operation again for the generation of a new spectrum $\underset{\sim}{S}''$ from $\underset{\sim}{S}'$, where $R_2'' \Leftrightarrow R_{23}'$

R_0''	R_1''	R_2''	R_3''	R_4''	R_{12}''	R_{13}''	R_{14}''
6	-6	6	-6	-2	2	-2	-6
R_{23}''	R_{24}''	R_{34}''	R_{123}''	R_{124}''	R_{134}''	R_{234}''	R_{1234}''
2	-2	2	-2	2	-2	-6	-2

see Figure 1.11.c.

Finally, applying the operation for the generation of a new spectrum $\underset{\sim}{S}'''$ from $\underset{\sim}{S}''$, where $R_4''' \Leftrightarrow R_{14}''$:

R_0'''	R_1'''	R_2'''	R_3'''	R_4'''	R_{12}'''	R_{13}'''	R_{14}'''
6	-6	6	-6	-6	2	-2	-2
R_{23}'''	R_{24}'''	R_{34}'''	R_{123}'''	R_{124}'''	R_{134}'''	R_{234}'''	R_{1234}'''
2	2	-2	-2	-2	2	-2	-6

see Figure 1.11.d.

Now if the first (n + 1) spectral coeffecients of this function are ordered by magnitude and rendered positive ,the result is:

6 6 6 6 6

which appears in the look-up tables of positive-characteristic-vectors (Appendix B);thus it is a threshold function .The threshold gate parameters may now be calculated using the method described above,namely the coeffecients:

R_0	R_1	R_2	R_3	R_4
6	-6	6	-6	-6,

give the corresponding weights:

$$\begin{array}{ccccc} w_0' & w_1' & w_2' & w_3' & w_4' \\ 1 & -1 & 1 & -1 & -1 \end{array}$$ see Appendix B.

From equation 1.26,the output weight is

$$\frac{1}{2}(\ 4 + 1 + 1\ ) = 3\ .$$

The resulting gate appears in Figure 1.11.e,together with the exclusive-Or circuitry necessary to carry out the spectral translations,i.e. x_1 replaced by $x_1 \oplus x_2$,x_2 replaced by $x_2 \oplus x_3$ and finally x_4 replaced by $x_1 \oplus x_4$. Because w_1' , w_3' , w_4' are negative,inverters are placed on the input lines x_1 , x_3 , x_4 before the threshold gate.

This example illustrates a property common to many non-threshold Boolean functions,that is such functions may be rendered linearly-seperable (threshold functions) by the application of the operation of spectral translation.Such functions will be said to have a threshold "embedded" within them.

The importance of this result of course lies in the fact that the versatility of threshold logic is increased many-fold by the straight-forward appending of equivalence (exclusive-OR) type logic. In fact ,the tables of Appendix C shows that there are only three classes of functions out of eighteen for $n \leqslant 4$ which do not have embedded threshold functions.These fifteen classes of embedded threshold functions represent 70 % of all $n \leqslant 4$ functions.

This spectral translation design method is to be combined with minterm-interchange design method in Chapter three, and the "cost" of the resulting method of synthesis will be considered,

REFERENCES

1. DERTOUZOS ,M.L. "Threshold Logic: A Synthesis Approach" Research Monograph No. 32,The M.I.T. Press,Cambridge,Massachusetts 1965.

2. LECHNER,R.J. "Harmonic Analysis of Switching Functions" in Recent Developments in Switching Theory, A Mukhopadyay,Ed.,New York, Academic Press 1971.

3. KARPOVSKY,M.G. "Finite Orthogonal Series in the Design of Digital Devices" John Wiley & Sons, Israel Universities Press,1976.

4. HURST,S.L. "The Logical Processing of Digital Signals" Edward Arnold ,London 1978.

5. RADEMACHER,H. "Einige Sötze Über Reihen von Allgonienen Orthogonal Functionen" Math. Ann. Vol.87 p.p.112-138, 1922.

6. WALSH,J.L. "A Closed Set of Normal Orthogonal Functions" Amer.J.Math. Vol.45 p.p. 5-24, 1923.

7. DAVIES, A.C. " Some Basic Ideas about Binary Discrete Signals" Symp.Theory and Applicafions of Walsh Functions,Hatfield(U.K)1971.

8. PALEY,R.E.A.C. "A Remarkable Series of Orthogonal Functions" Proc.Lond.Math.Soc. Vol.34, p.p. 241-279,1931.

9. CHOW,C.K. " On the Characterisation of Threshold Functions" I.E.E.E.Proc. Symp. Switching Teory and Logic Design,p.p.34-38,1961

10. LEWIS,P.M. and COATES,C.L. "Threshold Logic" John Wiley & Sons,1967

11. SHENG,C.L. "Threshold Logic" The Ryerson Press.Toronto,1969.

12. MUROGA ,S. "Threshold Logic and its Applications"John Wiley&Sons 1971.

13. ITO,T"Applications of the Walsh Functions to Pattern Recognition and Switching Theory"Proc,Symp,Application of Walsh Functions Naval Research Lab,U.S.A. 1970.

14. HURST,S.L. "The Applications of Chow Parameters and Rademacher-Walsh Matrices in the Synthesis of Binary Functions"Comp.J.Vol 16 No.2,1973.

15. EDWARDS,C.R. "The Application of Rademacher-Walsh Transform to Boolean Function Classification and Threshold Logic Synthesis"

I.E.E.E. Trans. on Compt. Vol. C-24,No.1 p.p.48-62,Jan.1975.

16. AHMED,N.,SCHREIBER,H.H.,and LOPRESTI,P.V."On the Notation and Definition of Terms Related to a class of Orthogonal Functions" I.E.E.E. Trans. EMC-15 p.p.75-80,1973.

17. HARMUTH,H.F. "Transmission of information by Orthogonal Functions" Spinger,New York,1969.

18. Proc.Symp. on Theory and Applications of Walsh Functions, Hatfield,U.K. June 1971.

19. Proc.Symp. on Theory and Applications of Walsh Functions and Nonsunusoidal Functions, Hatfield, U.K. June 1973.

20. HAAR,A. " Zur Theoire der Orthogonalen Functioneasysteme" Math. Annen 69, p.p. 331-371, 1910.

21. EDWARDS,C.R. "Matrix Methods in Combinational Logic Design" Ph.D. Thesis Bath University,England 1973.

22. WINDER,R.O. " Threshold Functions through n = 7 " Scientific Report No.7 R.C.A. Labs.Princeton,N.J. 1964.

23. LEWIS, P.M. " Practical Guide to Threshold Logic" Electron Design Vol.22 p.p. 66-88, 1967.

CHAPTER 2.A MINTERM INTERCHANGE OPERATION IN THE WALSH DOMAIN

CHAPTER 2 A MINTERM INTERCHANGE OPERATION IN THE WALSH DOMAIN

In this Chapter we will be considering the minterm interchange operation which is the corner stone of this thesis. First we will start with the relationships between Walsh spectra of Boolean functions which will be useful for minterm interchange operations.

2.1. Relationships Between Walsh Spectra of Boolean Functions

The basic operations defined in Boolean Algebra are Product, Sum, and Not. The Not operation has already been examined in the Walsh domain[1] (see Chapter 1.2.2.). Boolean Product and Sum operations[2] are considered below. The question we wish to answer is "Given the spectra of two Boolean functions f_1, f_2, what is the spectrum of their product ($\underset{\sim}{f}_p = \underset{\sim}{f}_1 \wedge \underset{\sim}{f}_2$), and Sum ($\underset{\sim}{f}_s = \underset{\sim}{f}_1 \vee \underset{\sim}{f}_2$) ? ".

2.1.1. Boolean Product (AND)

Theorem 2.1. The relationship between the product-function spectrum $\underset{\sim}{S}_p$ and the individual spectra $\underset{\sim}{S}_1$ and $\underset{\sim}{S}_2$ of the functions f_1 and f_2 is given by:

$$\underset{\sim}{S}_p = T\left[\text{diag.}F_1\right]T^{-1}\underset{\sim}{S}_2 + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}T\underset{\sim}{1} \qquad \text{...2.1.a}$$

$$\equiv T\left[\text{diag.}F_2\right]T^{-1}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 + \frac{1}{2}T\underset{\sim}{1} \qquad \text{...2.1.b.}$$

Proof:

The spectra of the product function f_p and the individual functions f_1, f_2 are given by definition as follows:

$$\underset{\sim}{S}_p \triangleq T\underset{\sim}{F}_p,$$

$$\underset{\sim}{S}_1 \triangleq T\underset{\sim}{F}_1,$$

$$\underset{\sim}{S}_2 \triangleq T\underset{\sim}{F}_2.$$

Using the linear relationships between the $\langle 0,1\rangle$ and $\langle 1,-1\rangle$ domains:

$$\underset{\sim}{F} = -2\,\underset{\sim}{f} + \underset{\sim}{1} \qquad \text{...2.2.a}$$

$$\underset{\sim}{f} = -\frac{1}{2}\underset{\sim}{F} + \frac{1}{2}\underset{\sim}{1} \qquad \text{...2.2.b}$$

$$\underset{\sim}{f}_p = \underset{\sim}{f}_1 \wedge \underset{\sim}{f}_2 = \left[\text{diag}.f_1\right]\underset{\sim}{f}_2 = \left[\text{diag}.f_2\right]\underset{\sim}{f}_1 \quad .$$

Therefore

$$\underset{\sim}{F}_p = -2\left[\text{diag}.f_1\right]\left(-\frac{1}{2}\underset{\sim}{F}_2 + \frac{1}{2}\underset{\sim}{1}\right) + \underset{\sim}{1}$$

and

$$\underset{\sim}{S}_p \overset{\Delta}{=} T\,\underset{\sim}{F}_p$$

$$= -2\,T\left\{\left[\text{diag}.f_1\right]\left(-\frac{1}{2}T^{-1}\underset{\sim}{S}_2 + \frac{1}{2}\underset{\sim}{1}\right)\right\} + T\,\underset{\sim}{1}$$

Therefore

$$\underset{\sim}{S}_p = T\left\{\left[\text{diag}.\ f_1\right]T^{-1}\underset{\sim}{S}_2 - T\,\underset{\sim}{f}_1\right\} + T\,\underset{\sim}{1}$$

But

$$-T\,\underset{\sim}{f}_1 + T\,\underset{\sim}{1} = -T\left(-\frac{1}{2}T^{-1}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{1}\right) + T\,\underset{\sim}{1}$$

$$= \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}T\,\underset{\sim}{1}$$

Therefore

$$\underset{\sim}{S}_p = T\left[\text{diag}.\ f_1\right]T^{-1}\underset{\sim}{S}_2 + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}T\,\underset{\sim}{1}$$

QED.

Interchaging the designations in the above will correspondingly yield the proof of the equation 2.1.b.

Example 2.1. Given:

$$\underset{\sim}{f}_1 = \begin{bmatrix} 1\\0\\1\\1\\0\\0\\0\\1 \end{bmatrix} \quad \text{and} \quad \underset{\sim}{f}_2 = \begin{bmatrix} 0\\1\\0\\1\\1\\1\\0\\1 \end{bmatrix}$$

$f_1=(\bar{x}_1\wedge\bar{x}_3)\vee(x_2\wedge x_3)$ $\qquad$ $f_2=x_3\vee(x_1\wedge\bar{x}_2)$

then working entirely in the conventional Boolean domain we have the AND relationship between f_1 and f_2 as follows;

$$\underset{\sim}{f}_p = \underset{\sim}{f}_1 \wedge \underset{\sim}{f}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad f_p = f_1 \wedge f_2 = x_2 \wedge x_3$$

The spectra of these functions are (see Chapter 1.2.1.)

$$\underset{\sim}{S}_1 = \begin{bmatrix} 0 \\ -4 \\ 4 \\ 0 \\ 0 \\ -4 \\ -4 \\ 0 \end{bmatrix} \qquad \underset{\sim}{S}_2 = \begin{bmatrix} -2 \\ 2 \\ -2 \\ 6 \\ 2 \\ 2 \\ -2 \\ 2 \end{bmatrix} \qquad \underset{\sim}{S}_p = \begin{bmatrix} 4 \\ 0 \\ 4 \\ 4 \\ 0 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

Evaluating $\underset{\sim}{S}_p$ using ,for example, equation 2.1.b,we have:

$$\underset{\sim}{S}_p = T \ \text{diag.}\left[(0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1)\right] \frac{1}{8} T^t \underset{\sim}{S}_1 + \frac{1}{2} \underset{\sim}{S}_2 + \frac{1}{2} T \underset{\sim}{1}$$

$$= T \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 1 \\ 1 \\ 0 \\ -1 \end{bmatrix} + \frac{1}{2} \underbrace{\begin{bmatrix} -2 \\ 2 \\ -2 \\ 6 \\ 2 \\ 2 \\ -2 \\ 2 \end{bmatrix}}_{\underset{\sim}{S}_2} + \frac{1}{2} \underbrace{\begin{bmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{T \underset{\sim}{1}}$$

$$= \begin{bmatrix} 1 \\ -1 \\ 5 \\ 1 \\ -1 \\ -1 \\ -3 \\ -1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ -1 \\ 3 \\ 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 4 \\ 4 \\ 0 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

,which is as previously determined.

The normal commutative and associative properties clearly enables this relation to be extended to more than two functions.

2.1.2. Boolean Sum (OR)

Theorem 2.2. The relationship between the Sum function f_s (spectrum S_s)and the individual functions f_1, f_2 (spectra S_1, S_2) is given by:

$$\underset{\sim}{S}_s = T\left[\text{diag.}\bar{f}_1\right]T^{-1}\underset{\sim}{S}_2 + \frac{1}{2}\underset{\sim}{S}_1 - \frac{1}{2}T\underset{\sim}{1} \qquad \ldots 2.3.a$$

$$\equiv T\left[\text{diag.}\bar{f}_2\right]T^{-1}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 - \frac{1}{2}T\underset{\sim}{1} \qquad \ldots 2.3.b.$$

Proof:

Since for any given function f: $\underset{\sim}{S}_f = -\underset{\sim}{S}_{\bar{f}}$ (see Chapter 1.2.2) and $f_s = f_1 \vee f_2 = \overline{(\bar{f}_1 \wedge \bar{f}_2)}$,from the equation 2.1.a we obtain :

$$\underset{\sim}{S}_s = -\left\{T\left[\text{diag.}\ \bar{f}_1\right]T^{-1}\underset{\sim}{S}_{\bar{f}_2} + \frac{1}{2}\underset{\sim}{S}_{\bar{f}_1} + \frac{1}{2}T\underset{\sim}{1}\right\}$$

$$= T\left[\text{diag.}\ \bar{f}_1\right]T^{-1}\underset{\sim}{S}_2 + \frac{1}{2}\underset{\sim}{S}_1 - \frac{1}{2}T\underset{\sim}{1}$$

QED.

Proof of equation 2.3.b follows similarly.

The commutative and associative properties enables this relation to be extended to more than two functions.

2.1.3. Development to Include Exclusive-OR Relations

Theorem 2.4. The relationship between spectral coeffecients $\bar{S}$ includes the following:

$$\underset{\sim}{S}_p = -\frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 + \frac{1}{2}T\underset{\sim}{1} \qquad \ldots 2.4.a$$

$$\underset{\sim}{S}_s = \frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 - \frac{1}{2}T\underset{\sim}{1} \qquad \ldots 2.4.b$$

where S_e is the spectrum of the exclusive-OR function $f_1 \oplus f_2$.

Proof :

$$\begin{aligned}
\underset{\sim}{S}_p &= T\underset{\sim}{F}_p \\
&= T\left\{-2\left[\text{diag. } f_1\right]\underset{\sim}{f}_2 + \underset{\sim}{1}\right\} \\
&= T\left\{-2(-\frac{1}{2}\left[\text{diag. } F_1\right] + \frac{1}{2}I)(-\frac{1}{2}\underset{\sim}{F}_2 + \frac{1}{2}\underset{\sim}{1}) + \underset{\sim}{1}\right\} \\
&\qquad\qquad (\text{ I= Unit matrix}) \\
&= T\left\{\frac{1}{2}(-\left[\text{diag. } F_1\right]\underset{\sim}{F}_2 + \left[\text{diag. } F_1\right]\underset{\sim}{1} + I\underset{\sim}{F}_2 - \underset{\sim}{1}) + \underset{\sim}{1}\right\} \\
&= \frac{1}{2}T\left\{-\left[\text{diag. } F_1\right]\underset{\sim}{F}_2 + \underset{\sim}{F}_1 + \underset{\sim}{F}_2 - \underset{\sim}{1}\right\} + T\underset{\sim}{1} \\
&= -\frac{1}{2}T\left[\text{diag. } F_1\right]\underset{\sim}{F}_2 + \frac{1}{2}T\underset{\sim}{F}_1 + \frac{1}{2}T\underset{\sim}{F}_2 + \frac{1}{2}T\underset{\sim}{1}\,.
\end{aligned}$$

It is known that[1,4] multiplication of elements of F_1, F_2 is equivalent to the exclusive-OR of elements of f_1, f_2 i.e $\left[\text{diag.} F_1\right]F_2$ in $\langle 1, -1\rangle$ domain corresponds $f_1 \oplus f_2$ in $\langle 0,1\rangle$ domain. Thence

$$\underset{\sim}{S}_p = -\frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 + \frac{1}{2}T\underset{\sim}{1}\,.$$

Further, using $f_1 \vee f_2 = (\overline{\underset{\sim}{\bar{f}}_1 \wedge \underset{\sim}{\bar{f}}_2})$ and $\underset{\sim}{S}_f = -\underset{\sim}{S}_{\bar{f}}$ and equation 2.4.a, we obtain

$$\underset{\sim}{S}_s = \frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 - \frac{1}{2}T\underset{\sim}{1}$$

since $f_1 \oplus f_2 = \bar{f}_1 \oplus \bar{f}_2$.

QED.

Example 2.2.

Given f_1 and f_2 as in example 2.1

$$f_e = f_1 \oplus f_2 = \bar{x}_2 \vee (\bar{x}_1 \wedge \bar{x}_3) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad F_e = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}, \quad S_e = \begin{bmatrix} -2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ 2 \\ 2 \end{bmatrix}$$

Using say equation 2.4.a, we may now complete the product spectrum:

$$\underset{\sim}{S}_p = -\frac{1}{2} \begin{bmatrix} -2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ 2 \\ 2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ -4 \\ 4 \\ 0 \\ 0 \\ -4 \\ -4 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -2 \\ 2 \\ -2 \\ 6 \\ 2 \\ 2 \\ -2 \\ 2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 4 \\ 4 \\ 0 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

which is the same result for S_p found in example 1.2.

Corollary 2.1. If f_1 and f_2 are disjoint(i.e. $f_1 \oplus f_2 = f_1 + f_2$) then $\underset{\sim}{S}_s = \underset{\sim}{S}_1 + \underset{\sim}{S}_2 - T\,\underset{\sim}{1}$.

2.1.4. Bolean Product (AND) Boolean Sum (OR) for Three Functions

Theorem 2.4. The relationship between the Product-function spectrum S_p and the individual spectra S_1, S_2, and S_3 is given by :

$$\underset{\sim}{S}_p = \frac{1}{4}\left\{\underset{\sim}{S}_{e_{123}} - (\underset{\sim}{S}_{e_{12}} + \underset{\sim}{S}_{e_{13}} + \underset{\sim}{S}_{e_{23}}) + (\underset{\sim}{S}_1 + \underset{\sim}{S}_2 + \underset{\sim}{S}_3)\right\} + \frac{3}{4} T \underset{\sim}{1} \quad \ldots 2.5$$

where $S_{e_{12}}$ is the spectrum of the function $f_1 \oplus f_2$ and $S_{e_{123}}$ is the spectrum of the function $f_1 \oplus f_2 \oplus f_3$ etc.

Proof:

$$\underset{\sim}{f}_p = \underset{\sim}{f}_1 \wedge \underset{\sim}{f}_2 \wedge \underset{\sim}{f}_3 = \left[\text{diag.} f_1\right]\left[\text{diag.} f_2\right]\underset{\sim}{f}_3$$

Writing the Product function in $\langle -1,1\rangle$ domain and evaluating its spectrum gives

$$\underset{\sim}{S}_p = T\left\{-2\left(\left[\text{diag.} f_1\right]\left[\text{diag.} f_2\right]\underset{\sim}{f}_3\right) + \underset{\sim}{1}\right\}$$

$$= -2\,T\left\{\left(-\frac{1}{2}\left[\text{diag.} F_1\right] + \frac{1}{2} I\right)\left(-\frac{1}{2}\left[\text{diag.} F_2\right] + \frac{1}{2} I\right)\left(-\frac{1}{2}\underset{\sim}{F}_3 + \frac{1}{2}\underset{\sim}{1}\right)\right\} + T\underset{\sim}{1}$$

$$= -\frac{1}{2^2}T\left\{-\left[\text{diag.} F_1\right]\left[\text{diag.} F_2\right]\underset{\sim}{F}_3 + \left[\text{diag.} F_1\right]\underset{\sim}{F}_3 + \left[\text{diag.} F_2\right]\underset{\sim}{F}_3 + \left[\text{diag.} F_1\right]\left[\text{diag.} F_2\right]\underset{\sim}{1} - \left[\text{diag.} F_1\right]\underset{\sim}{1} - \left[\text{diag.} F_2\right]\underset{\sim}{1} - \underset{\sim}{F}_3 + \underset{\sim}{1}\right\} + T\underset{\sim}{1}$$

Since $\left[\text{diag } F_1\right]\left[\text{diag.} F_2\right]\underset{\sim}{F}_3$ corresponds to $f_1 \oplus f_2 \oplus f_3$ in $\langle 0,1\rangle$ domain and similiarly $\left[\text{diag.} F_1\right]\underset{\sim}{F}_2$ to $f_1 \oplus f_2$ in $\langle 0,1\rangle$ domain etc., then

$$\underset{\sim}{S}_p = \frac{1}{4}\underset{\sim}{S}_{e_{123}} - \frac{1}{4}(\underset{\sim}{S}_{e_{12}} + \underset{\sim}{S}_{e_{13}} + \underset{\sim}{S}_{e_{23}}) + \frac{1}{4}(\underset{\sim}{S}_1 + \underset{\sim}{S}_2 + \underset{\sim}{S}_3) + \frac{3}{4}T\underset{\sim}{1}$$

QED.

Theorem 2.5 The relationship between the Sum-function spectrum S_s and the individual functions spectra S_1, S_2, and S_3 is given by:

$$\underset{\sim}{S}_s = \frac{1}{4}\left\{\underset{\sim}{S}_{e_{123}} + (\underset{\sim}{S}_{e_{12}} + \underset{\sim}{S}_{e_{13}} + \underset{\sim}{S}_{e_{23}}) + (\underset{\sim}{S}_1 + \underset{\sim}{S}_2 + \underset{\sim}{S}_3)\right\} - \frac{3}{4} T \underset{\sim}{1} \quad \ldots 2.6$$

Where $S_{e_{123}}$ is the spectrum of the function $f_1 \oplus f_2 \oplus f_3$ etc.

Proof:

Since $f_1 \vee f_2 \vee f_3 = \overline{(\bar{f}_1 \wedge \bar{f}_2 \wedge \bar{f}_3)}$

$\bar{f}_1 \oplus \bar{f}_2 \oplus \bar{f}_3 = f_1 \oplus f_2 \oplus f_3$

$\bar{f}_1 \oplus \bar{f}_2 = f_1 \oplus f_2$

and $S_f = - S_{\bar{f}}$,

then using the equation 2.5 ,we obtain

$$S_s = - S_{\bar{f}_1 \wedge \bar{f}_2 \wedge \bar{f}_3}$$

$$= \frac{1}{4}\left\{\underset{\sim}{S}_{e_{123}} + (\underset{\sim}{S}_{e_{12}} + \underset{\sim}{S}_{e_{13}} + \underset{\sim}{S}_{e_{23}}) + (\underset{\sim}{S}_1 + \underset{\sim}{S}_2 + \underset{\sim}{S}_3)\right\} - \frac{3}{4} T \underset{\sim}{1}.$$

QED.

Example 2.3.

$$\underset{\sim}{f}_1 \triangleq \begin{bmatrix}1\\1\\0\\0\\1\\0\\0\\0\end{bmatrix}, \quad \underset{\sim}{f}_2 \triangleq \begin{bmatrix}1\\0\\0\\1\\0\\0\\1\\1\end{bmatrix}, \quad \underset{\sim}{f}_3 \triangleq \begin{bmatrix}1\\1\\0\\0\\1\\0\\1\\1\end{bmatrix}, \quad ; \therefore \underset{\sim}{f}_s = \underset{\sim}{f}_1 \vee \underset{\sim}{f}_2 \vee \underset{\sim}{f}_3 = \begin{bmatrix}1\\1\\0\\1\\1\\0\\1\\1\end{bmatrix}$$

$f_1 = (x_1 \wedge x_2) \vee (\bar{x}_2 \wedge \bar{x}_3)$

$f_2 = (x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (\bar{x}_1 \wedge \bar{x}_2 \wedge x_3).$

$f_3 = (\bar{x}_1 \wedge \bar{x}_2) \vee (x_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_3)$

$f_s = \overline{(\bar{x}_1 \wedge x_2 \wedge \bar{x}_3) \vee (x_1 \wedge \bar{x}_2 \wedge x_3)}$

$$\underset{\sim}{f}_{e_{12}} = \begin{bmatrix}0\\1\\0\\1\\1\\0\\1\\1\end{bmatrix}, \quad \underset{\sim}{f}_{e_{13}} = \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\1\end{bmatrix}, \quad \underset{\sim}{f}_{e_{23}} = \begin{bmatrix}0\\1\\0\\1\\1\\0\\0\\0\end{bmatrix}, \quad \underset{\sim}{f}_{e_{123}} = \begin{bmatrix}1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$$

$$f_{e_{12}} \triangleq f_1 \oplus f_2 = (x_1 \wedge \bar{x}_3) \vee (x_2 \wedge x_3) \vee (\bar{x}_1 \wedge x_3)$$

$$f_{e_{13}} \triangleq f_1 \oplus f_3 = x_1 \wedge x_2$$

$$f_{e_{23}} \triangleq f_2 \oplus f_3 = (\bar{x}_1 \wedge x_3) \vee (x_1 \wedge \bar{x}_2 \wedge \bar{x}_3)$$

$$f_{e_{123}} \triangleq f_1 \oplus f_2 \oplus f_3 = (\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3) \vee (\bar{x}_1 \wedge x_2 \wedge x_3)$$

$$\underset{\sim}{S}_1 = \begin{bmatrix} 2 \\ -2 \\ -6 \\ -2 \\ -2 \\ 2 \\ -2 \\ 2 \end{bmatrix} \quad \underset{\sim}{S}_2 = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ -4 \\ 0 \\ -4 \\ -4 \end{bmatrix} \quad \underset{\sim}{S}_3 = \begin{bmatrix} -2 \\ 2 \\ -2 \\ -2 \\ -6 \\ 2 \\ -2 \\ 2 \end{bmatrix} \quad \underset{\sim}{S}_s = \begin{bmatrix} -4 \\ 0 \\ 0 \\ 0 \\ -4 \\ 4 \\ -4 \\ 0 \end{bmatrix}$$

$$\underset{\sim}{S}_{e_{12}} = \begin{bmatrix} -2 \\ 2 \\ 2 \\ 2 \\ -2 \\ 6 \\ -2 \\ 2 \end{bmatrix} \quad \underset{\sim}{S}_{e_{13}} = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 0 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \underset{\sim}{S}_{e_{23}} = \begin{bmatrix} 2 \\ -2 \\ -2 \\ 2 \\ 2 \\ 6 \\ -2 \\ 2 \end{bmatrix} \quad \underset{\sim}{S}_{e_{123}} = \begin{bmatrix} 4 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4 \\ -4 \end{bmatrix}$$

Now from equation 2.6 we can confirm the above Sum spectrum:

$$\frac{1}{4}\left\{ \underbrace{\begin{bmatrix} 4 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4 \\ -4 \end{bmatrix}}_{\underset{\sim}{S}_{e_{123}}} + \underbrace{\begin{bmatrix} 4 \\ 4 \\ 4 \\ 0 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\underset{\sim}{S}_{e_{13}}} + \underbrace{\begin{bmatrix} -2 \\ 2 \\ 2 \\ 2 \\ -2 \\ 6 \\ -2 \\ 2 \end{bmatrix}}_{\underset{\sim}{S}_{e_{12}}} + \underbrace{\begin{bmatrix} 2 \\ -2 \\ -2 \\ 2 \\ 2 \\ 6 \\ -2 \\ 2 \end{bmatrix}}_{\underset{\sim}{S}_{e_{23}}} + \underbrace{\begin{bmatrix} 2 \\ -2 \\ -6 \\ -2 \\ -2 \\ 2 \\ -2 \\ 2 \end{bmatrix}}_{\underset{\sim}{S}_1} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ -4 \\ 0 \\ -4 \\ -4 \end{bmatrix}}_{\underset{\sim}{S}_2} + \underbrace{\begin{bmatrix} -2 \\ 2 \\ -2 \\ -2 \\ -6 \\ 2 \\ -2 \\ 2 \end{bmatrix}}_{\underset{\sim}{S}_3} \right\} - \frac{3}{4} \underbrace{\begin{bmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{T\,\underset{\sim}{1}} = \underbrace{\begin{bmatrix} -4 \\ 0 \\ 0 \\ 0 \\ -4 \\ 4 \\ -4 \\ 0 \end{bmatrix}}_{\underset{\sim}{S}_s}$$

Theorem 2.6 The relationship between the spectra of the Product-function S_p, the Sum-function S_s, and the spectra of individual functions S_1 and S_2 is given by:

$$\underset{\sim}{S}_p + \underset{\sim}{S}_s = \underset{\sim}{S}_1 + \underset{\sim}{S}_2 \qquad \ldots 2.7$$

Proof:

Adding equations 2.4.a and 2.4.b

$$\begin{aligned} \underset{\sim}{S}_p + \underset{\sim}{S}_s &= -\frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 + \frac{1}{2}T\underset{\sim}{1} \\ &\quad + \frac{1}{2}\underset{\sim}{S}_e + \frac{1}{2}\underset{\sim}{S}_1 + \frac{1}{2}\underset{\sim}{S}_2 - \frac{1}{2}T\underset{\sim}{1} \\ &= \underset{\sim}{S}_1 + \underset{\sim}{S}_2 \end{aligned}$$

QED.

2.2. The Minterm Interchange Operation in the Walsh Domain

2.2.1 General Formulation of the Minterm Interchange in the Walsh Domain

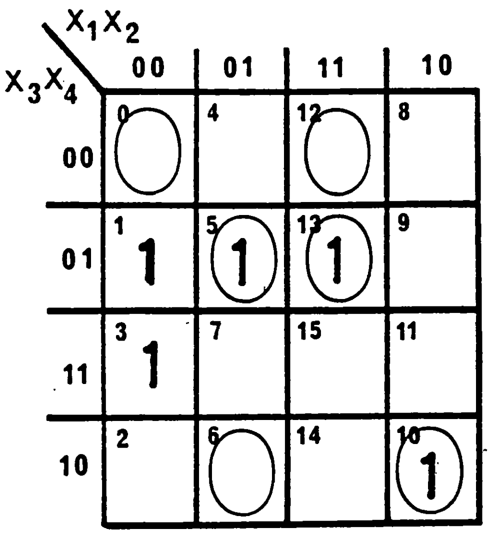
The interchange of one true minterm with another, or one false minterm with another, does not modify the function. We therefore consider only the interchange of true minterms with false ones [3].

Definition 2.1 " Interchanged Function (δ)" is the function which we find after having changed one or more of the true-minterms of a Boolean function f with false ones.

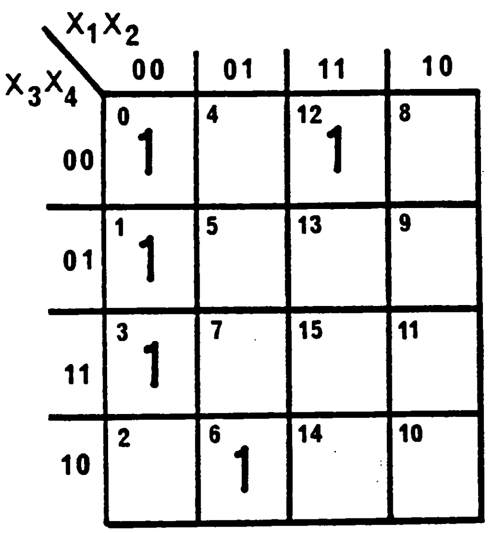
Definition 2.2 " True function (α) " is composed of the true minterms which are to be changed.

Definition 2.3 "False Function (β) " is composed of the false minterms which are to be changed.

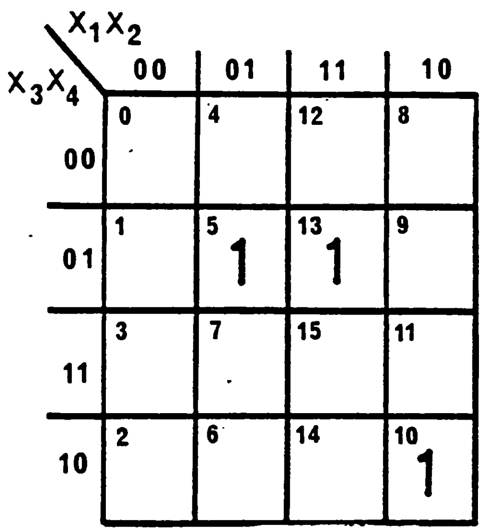
It is obvious that the α and β functions are disjoint.



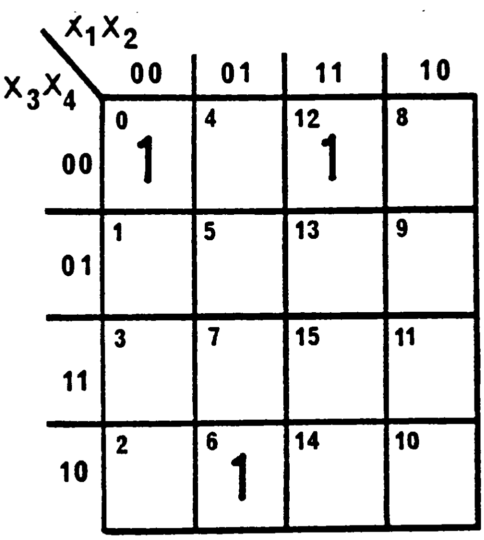
a)Example Boolean Function f.



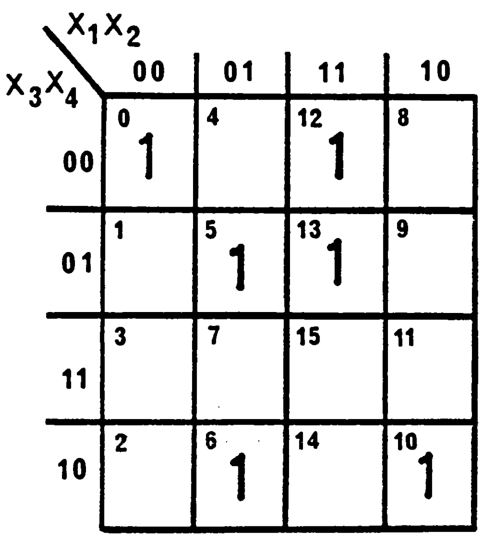
b) Interchaged function δ



c)True function α



d) False function β



e) Changer function γ

Figure 2.1. Minterm interchage on Boolean function f and related functions

Definition 2.4 "Changer Function (γ)" is the sum or exclusive-Or of disjoint α and β functions i.e.

$$\gamma = \alpha \vee \beta = \alpha \oplus \beta \qquad \ldots 2.8$$

Example 2.4 Let us consider interchanging the true minterms m_5, m_{13}, m_{10} of the Boolean function f in Figure 2.1 with any of the three false minterms m_0, m_6, m_{12} .The above-defined functions are illustrated in Figure 2.1

Corollary 2.2 From the above definitions and the example the relationship below can easily be observed;

$$f = \gamma \oplus \delta \qquad \ldots 2.9$$

$$\beta = \gamma \wedge \delta \qquad \ldots 2.10$$

Theorem 2.7 The relationship between the spectrum of the interchanged function, S_δ , and the spectrum of the original function S_f, together with the true function spectrum S_α and the false function spectrum S_β is given by:

$$\underset{\sim}{S}_\delta = \underset{\sim}{S}_f + \underset{\sim}{S}_\beta - \underset{\sim}{S}_\alpha \qquad \ldots 2.11$$

Proof: Choosing f_1 and f_2 as α and β in equation 2.4.b

$$\underset{\sim}{S}_{\alpha+\beta} = \frac{1}{2}\underset{\sim}{S}_{\alpha\oplus\beta} + \frac{1}{2}\underset{\sim}{S}_\alpha + \frac{1}{2}\underset{\sim}{S}_\beta - \frac{1}{2} T \underset{\sim}{1}$$

$\gamma = \alpha + \beta = \alpha \oplus \beta$ (equation 2.8),whence

$$\underset{\sim}{S}_\gamma = \underset{\sim}{S}_\alpha + \underset{\sim}{S}_\beta - T \underset{\sim}{1} \qquad \ldots 2.12$$

Considering equation 2.10 and substituting γ, δ and β for f_1 , f_2 and $f_1 \wedge f_2$ in the equation 2.4.a gives

$$\underset{\sim}{S}_\beta = -\frac{1}{2}\underset{\sim}{S}_{\gamma\oplus\delta} + \frac{1}{2}\underset{\sim}{S}_\gamma + \frac{1}{2}\underset{\sim}{S}_\delta + \frac{1}{2} T \underset{\sim}{1} \qquad \ldots 2.13$$

Using the equation 2.9 and 2.12 in the equation 2.13 and rearranging it, we find

$$\underset{\sim}{S}_\delta = \underset{\sim}{S}_f + \underset{\sim}{S}_\beta - \underset{\sim}{S}_\alpha \ .$$

QED.

Example 2.5 Let us apply the equation 2.11 for the previous example 2.4, see Figure 2.1 :

$$\underset{\sim f}{\begin{bmatrix} 6 \\ -2 \\ -2 \\ -2 \\ 6 \\ -2 \\ -2 \\ 6 \\ 6 \\ -2 \\ 6 \\ -2 \\ 6 \\ -2 \\ -2 \\ -2 \end{bmatrix}}_{S} + \underset{\sim \beta}{\begin{bmatrix} 10 \\ -2 \\ 2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ -2 \\ 2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ -6 \end{bmatrix}}_{S} - \underset{\sim \alpha}{\begin{bmatrix} 10 \\ 2 \\ 2 \\ -2 \\ 2 \\ 2 \\ -2 \\ 2 \\ 6 \\ -6 \\ 6 \\ -2 \\ 2 \\ -2 \\ -2 \\ -2 \end{bmatrix}}_{S} = \underset{\sim \delta}{\begin{bmatrix} 6 \\ -6 \\ -2 \\ -2 \\ -2 \\ -6 \\ 2 \\ 2 \\ -2 \\ 6 \\ -2 \\ -6 \\ 2 \\ 2 \\ -2 \\ -6 \end{bmatrix}}_{S}$$

$S_f \qquad S_\beta \qquad S_\alpha \qquad S_\delta$

2.2.2 Spectrum of the Interchanged Function S in terms of Rademacher-Walsh Functions

Definition 2.5 "Minterm function f_{m_i} " is the function which has one true-minterm value only, in position m_i.

Now, we can write the true and false functions in terms of the minterm functions as:

$$\alpha = \sum_{i=1}^{k} \underset{\sim}{f}_{m_i} \qquad 0 \leqslant m_i \leqslant 2^n-1 \qquad \ldots 2.14$$

$$\beta = \sum_{j=1}^{k} \underset{\sim}{f}_{m_j} \qquad 0 \leqslant m_j \leqslant 2^n-1 \qquad \ldots 2.15$$

where m_i s are the true minterms to be changed, m_js are the false minterms to be changed, and k is the number of the true (or false) minterms to be changed. Therefore

$$\underset{\sim}{S}_\beta = T\left\{-2 \sum_{j=1}^{k} \underset{\sim}{f}_{m_j} + \underset{\sim}{1}\right\}$$

$$= -2 \sum_{j=1}^{k} \underset{\sim}{t}_{m_j} + T \underset{\sim}{1} \qquad \ldots 2.16$$

where t_{m_j} is the m_j th column vector of the transform matrix T.

Similiarly

$$\underset{\sim}{S}_{*} = T \left\{ -2 \sum_{i=1}^{k} \underset{\sim}{f}_{m_i} + \underset{\sim}{1} \right\}$$

$$= -2 \sum_{i=1}^{k} t_{m_i} + T \underset{\sim}{1} \qquad \ldots 2.17$$

where t_{m_i} is the m_i th column vector of the transform matrix T.

Using the equations 2.16 and 2.17 in the equation 2.11 :

$$\underset{\sim}{S}_{\delta} = \underset{\sim}{S}_{f} + 2 \sum_{i=1}^{k} \underset{\sim}{t}_{m_i} - \sum_{j=1}^{k} \underset{\sim}{t}_{m_j}$$

or

$$= \underset{\sim}{S}_{f} + 2 \left\{ \sum_{i,j=1}^{k} (\underset{\sim}{t}_{m_i} - \underset{\sim}{t}_{m_j}) \right\} \qquad \ldots 2.18$$

If the transform matrix is in Hadamard order (T_H) then the t_m column vectors would be the Rademacher-Walsh functions corresponding to the same minterm . Thus the equation 2.18 shows the spectrum of the interchanged function S_{δ} in terms of Rademacher-Walsh functions.

Example 2.6

Let us consider the previous function f of example 2.4 firstly using the T matrix (in Rademacher-Walsh order) then secondly the T_H matrix in Hadamard order. This is shown below:

Rademacher-Walsh ordering:

$$\underset{\tilde{S}\delta}{\begin{bmatrix} 6\\-6\\-2\\-2\\-2\\-6\\2\\2\\-2\\6\\-2\\-6\\2\\2\\-2\\-6 \end{bmatrix}} = \underset{\tilde{S}}{\begin{bmatrix} 6\\-2\\-2\\-2\\6\\-2\\-2\\6\\6\\-2\\6\\-2\\6\\-2\\-2\\-2 \end{bmatrix}} + 2\left\{ \underset{\tilde{t}_5}{\begin{bmatrix} 1\\1\\-1\\1\\-1\\-1\\1\\-1\\-1\\1\\-1\\1\\-1\\1\\1\\1 \end{bmatrix}} + \underset{\tilde{t}_{10}}{\begin{bmatrix} 1\\-1\\1\\-1\\1\\-1\\1\\-1\\-1\\1\\-1\\1\\-1\\1\\-1\\1 \end{bmatrix}} + \underset{\tilde{t}_{13}}{\begin{bmatrix} 1\\-1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\1\\-1 \end{bmatrix}} - \underset{\tilde{t}_0}{\begin{bmatrix} 1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1 \end{bmatrix}} - \underset{\tilde{t}_6}{\begin{bmatrix} 1\\1\\-1\\-1\\1\\-1\\-1\\1\\1\\-1\\-1\\1\\-1\\-1\\1\\1 \end{bmatrix}} - \underset{\tilde{t}_{12}}{\begin{bmatrix} 1\\-1\\-1\\1\\1\\1\\-1\\-1\\-1\\-1\\1\\1\\1\\-1\\-1\\1 \end{bmatrix}} \right\}$$

Hadamard ordering:

$$\underset{\tilde{S}_\delta}{\begin{bmatrix} 6\\-2\\-2\\-2\\-2\\6\\-2\\-2\\-6\\2\\2\\2\\-6\\2\\-6\\-6 \end{bmatrix}} = \underset{\tilde{S}_f}{\begin{bmatrix} 6\\6\\-2\\6\\-2\\-2\\6\\-2\\-2\\6\\-2\\-2\\-2\\6\\-2\\-2 \end{bmatrix}} + 2\left\{ \underset{\tilde{W}_5}{\begin{bmatrix} 1\\-1\\1\\-1\\-1\\1\\-1\\1\\1\\-1\\1\\-1\\-1\\1\\-1\\1 \end{bmatrix}} + \underset{\tilde{W}_{10}}{\begin{bmatrix} 1\\1\\-1\\-1\\1\\1\\-1\\-1\\-1\\-1\\1\\1\\-1\\-1\\1\\1 \end{bmatrix}} + \underset{\tilde{W}_{13}}{\begin{bmatrix} 1\\-1\\1\\-1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\1\\-1\\1\\-1 \end{bmatrix}} - \underset{\tilde{W}_0}{\begin{bmatrix} 1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1 \end{bmatrix}} - \underset{\tilde{W}_6}{\begin{bmatrix} 1\\1\\-1\\-1\\-1\\-1\\1\\1\\1\\1\\-1\\-1\\-1\\-1\\1\\1 \end{bmatrix}} - \underset{\tilde{W}_{12}}{\begin{bmatrix} 1\\1\\1\\1\\-1\\-1\\-1\\-1\\-1\\-1\\-1\\-1\\1\\1\\1\\1 \end{bmatrix}} \right\}$$

Corresponding Boolean functions:

$$\tilde{W}_5 \uparrow \underbrace{}_{x_2 \oplus x_4} \quad \tilde{W}_{10} \uparrow \underbrace{}_{x_1 \oplus x_3} \quad \tilde{W}_{13} \uparrow \underbrace{}_{x_1 \oplus x_2 \oplus x_4} \quad \tilde{W}_0 \uparrow \underbrace{}_{x_0} \quad \tilde{W}_6 \uparrow \underbrace{}_{x_2 \oplus x_3} \quad \tilde{W}_{12} \uparrow \underbrace{}_{x_1 \oplus x_2}$$

2.2.3 The Effect of the Minterm-interchange on the First Order Spectral Coeffecients

Since we will be dealing with the first order spectral coeffecients of a Boolean function in the following two Chapters,it would be useful to choose the transform matrix in Rademacher-Walsh order rather than Hadamard order.This is because the first (n + 1) entries in the spectrum vector give the first order spectrum coeffecients including R_o when the Rademacher-Walsh order transform is used.

From Chapter 1 equation 1.4

$$\underset{\sim}{R}_s(x) = \underset{\sim}{W}_{2^s}(x) = (-\underset{\sim}{1})^{x_s(x)} \qquad 1 \leqslant s \leqslant n$$

where $R_s(x)$ = A Rademacher-Walsh function

$\underset{\sim}{W}_{2^s}(x)$= Particular rows(columns) of the Hadamard order transform matrix T_H which have index number as a integer power of 2.

$\underset{\sim}{x}_s(x)$ = s th independent variable

On the other hand ,since the entries of x_s are '0' or'1' then the above equation becomes

$$\underset{\sim}{R}_s(x) = -2\ \underset{\sim}{x}_s(x) + \underset{\sim}{1} \qquad ...2.19$$

where $\underset{\sim}{1} = [1\ \ 1\ \ldots .1]^t$.

Therefore

$$\underset{\sim}{X}_s(x) = \frac{1}{2}\left[-\underset{\sim}{R}_s(x) + \underset{\sim}{1}\right] \qquad ...2.20$$

Also

$$m_i = \underset{\sim}{x}(x) = \sum_{s=1}^{n} \underset{\sim}{x}_s\ 2^{n-s} \qquad ...2.21$$

where entries of x are the minterm identification numbers in decimal.

Substituting x_s in equation 2.21 for the x_s in the equation 2.20 gives

$$\underset{\sim}{x} = \sum_{s=1}^{n} \Big\{ \underbrace{\tfrac{1}{2}\left(-\underset{\sim}{R}_s(x) + \underset{\sim}{1}\right)}_{\text{Rademacher function in } 0,1 \text{ domain}} \Big\}\ 2^n \qquad ...2.22$$

Equation 2.22 shows that if we write the Rademacher functions as the rows of a matrix ,the columns of this matrix correspond to the binary expansions of the minterm identification numbers in$\langle 1,-1\rangle$ domain. This is illustrated by the following three variable example:

$$\begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \end{array}\begin{array}{c} \begin{array}{cccccccc} M_0 & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \end{array}$$

in$\langle 1,-1\rangle$ domain

$$\begin{array}{c} \\ 2^2 \\ 2^1 \\ 2^0 \\ \\ \end{array}\begin{array}{c} \begin{array}{cccccccc} m_0 & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ \begin{array}{cccccccc} (0) & (1) & (2) & (3) & (4) & (5) & (6) & (7) \end{array} \end{array}$$

in $\langle 0,1\rangle$ domain.

M_i denotes a column vector in $\langle 1,-1\rangle$domain, which corresponds to the binary expansion of the minterm identification number m_i in$\langle 0,1\rangle$domain.

Since we are dealing with the first order coeffecients only,we can write equation 2.18 as follows :

$$\underset{\sim}{S}_{\delta} = \underset{\sim}{S}_{f} + 2\left\{\sum_{i=1}^{k} \underset{\sim}{M}_i - \sum_{j=1}^{k} \underset{\sim}{M}_j\right\} \qquad \ldots 2.23$$

where M_i's are interchanged true minterms in binary $\langle 1\ -1\rangle$domain and M_j's are interchaged false minterms in binary $\langle 1,-1\rangle$ domain.

Example 2.7 Let us apply equation 2.23 to the previous example 2.4 :

$$\underset{\tilde{S}}{\begin{bmatrix}6\\-6\\-2\\-2\\-2\end{bmatrix}} = \underset{\tilde{S}_f}{\begin{bmatrix}6\\-2\\-2\\-2\\6\end{bmatrix}} +2\left\{ \underset{M_5}{\begin{bmatrix}1\\1\\-1\\1\\-1\end{bmatrix}} + \underset{M_{10}}{\begin{bmatrix}1\\-1\\1\\-1\\1\end{bmatrix}} + \underset{M_{13}}{\begin{bmatrix}1\\-1\\-1\\1\\-1\end{bmatrix}} - \underset{M_0}{\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}} - \underset{M_6}{\begin{bmatrix}1\\1\\-1\\-1\\1\end{bmatrix}} - \underset{M_7}{\begin{bmatrix}1\\-1\\-1\\1\\1\end{bmatrix}} \right\}$$

Corollary 2.3. The effect on the first order spectral cceffecients of changing a pair of minterms may be zero, -4 or +4.

This can easily be seen from equation 2.23.

Corollary 2.4 If two minterms ,having a Hamming distance of one,are interchanged then only one corresponding first_order spectral coeffecient will be effected, but the rest of the first_order spectral coeffecients will not.

Equation 2.23 confirms the above corollary.

2.3 Conclusion

In this Chapter we have mathematically shown the effect of minterm-interchange on the spectral coeffecients. The following two Chapters will consider the application of this minterm-interchange to the design of the combinational circuits and sequential machines. with the object of generating members of a class of simplest functions,which may then yield an optimal synthesis realisation.

REFERENCES

1. HURST ,S.L. "The Logical Processing of Digital Signals" Edward Arnold ,London,1978.

2. ERIS,E."Relationship between Rademacher-Walsh Spectra of Boolean Functions" IEE Computers and Digital Techniques (CDT) Vol.1 , p.p. 45-48,May 1978.

3. ERIS,E. " A Minterm Interchange Operation in the Walsh Domain " Electronics Letters,14,No.4 ,p.p.92-94,Feb.1978.

4. EDWARDS, C.R. " The Application of the Rademacher-Walsh Transform to Boolean Function Classification and Threshold Logic Synthesis" Trans. IEEE, C-24,p.p.48-62 ,Jan.1975.

CHAPTER 3. THE APPLICATION OF THE MINTERM-INTRECHANGE OPERATION TO THE DESIGN OF COMBINATIONAL LOGIC

Chapter 3 THE APPLICATION OF THE MINTERM-INTERCHANGE OPERATION TO THE DESIGN OF COMBINATIONAL LOGIC

3.1. Complexity of a Boolean Function and its Relationship to Minterm-interchange

3.1.1. Concept of Complexity

As is well known , many different methods have been used for synthesising logic circuits which realise Boolean functions.Take for example, the function f in Figure 3.1.Some of the circuits which realise this Boolean function are shown in Figure 3.2.

Because of these many different forms of realisations,we face the problem of choosing the 'best' circuit ,that is the circuit of minumum cost or complexity.In order to decide what constitues a minumum cost,or minimum complexity,the following parameters may be taken into account :

X_3X_4 \ X_1X_2	00	01	11	10
00		1	1	
01	1		1	
11		1		1
10	1		1	

Figure 3.1. The example

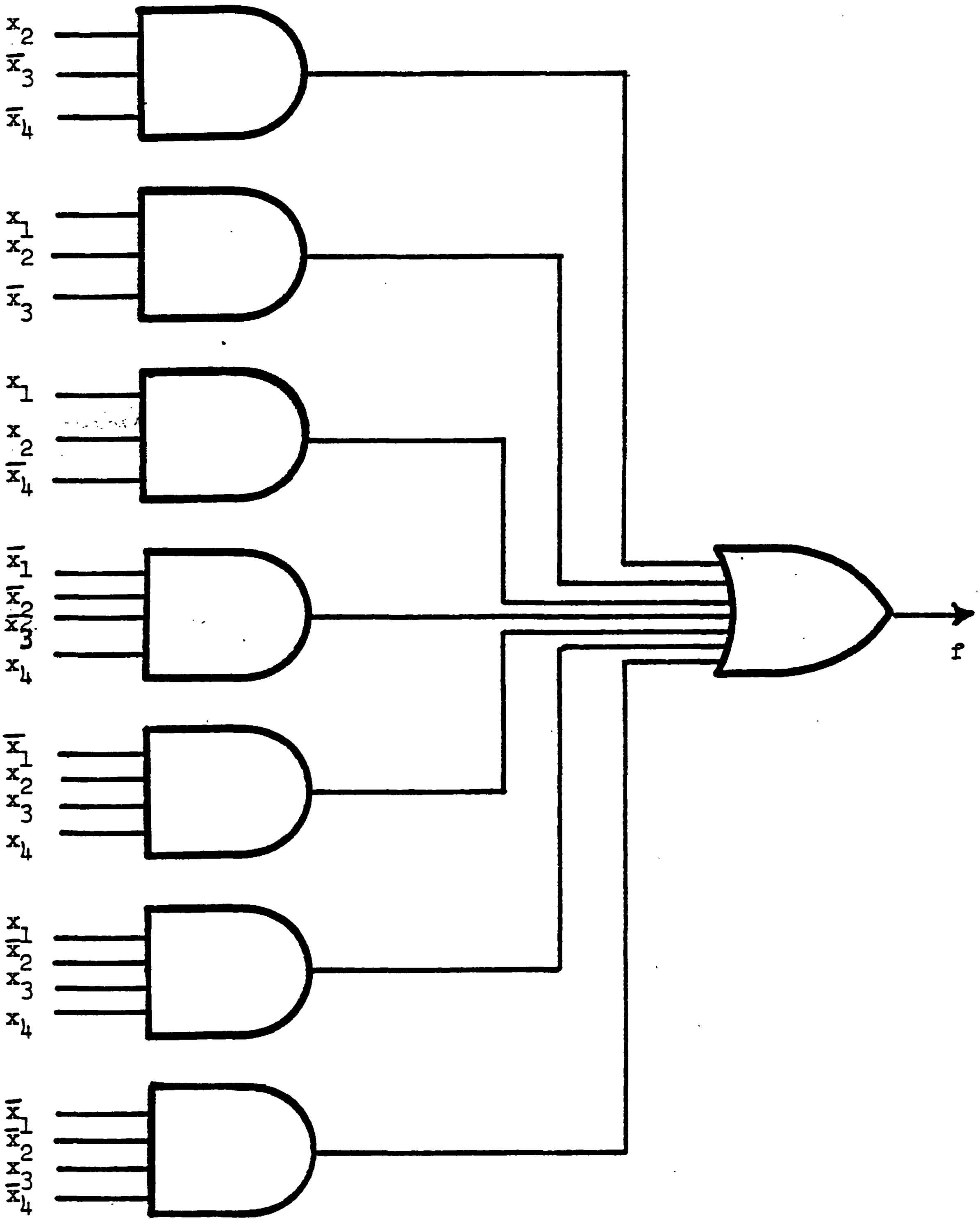


Figure 3.2.a Two-level realisation of the example function

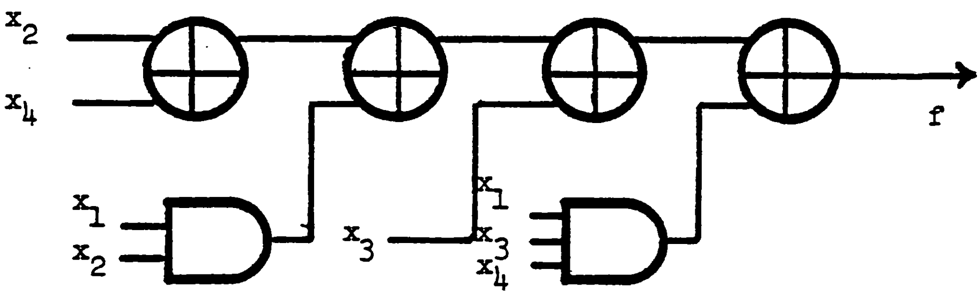


Figure 3.2.b Realisation of the example function by using the Reed-Muller expansion.

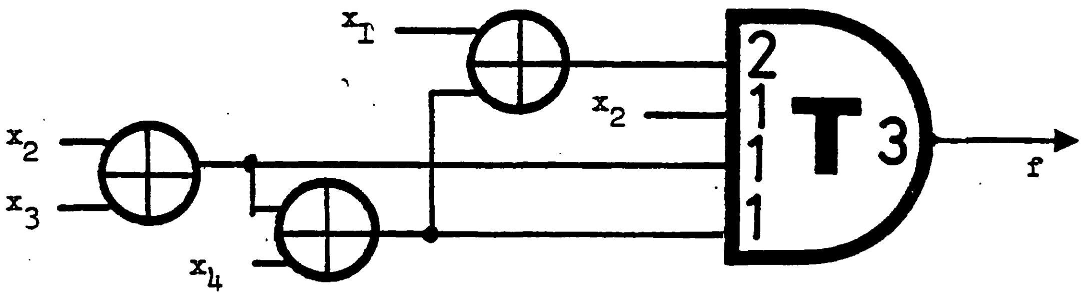


Figure 3.2.c Realisation of the example funtion by using Threshold gates.

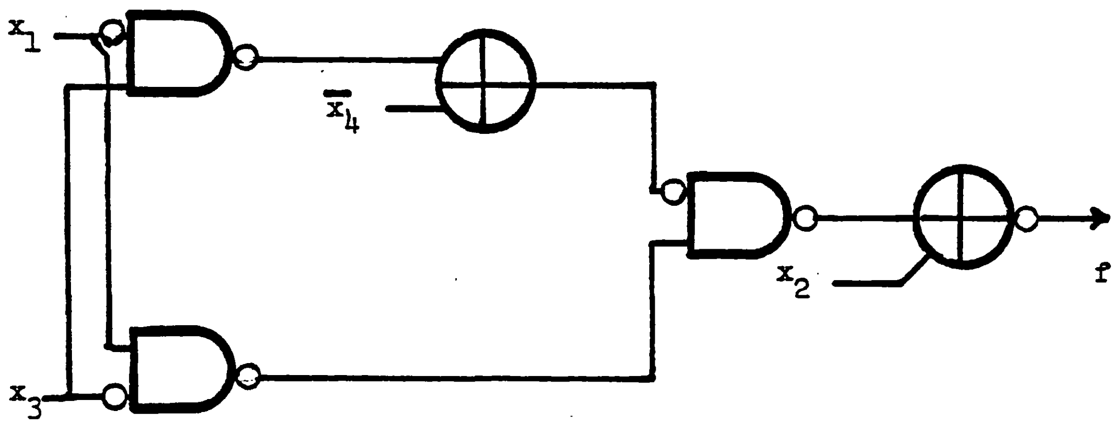


Figure 3.2.d. Symmetry realisation of the example function.

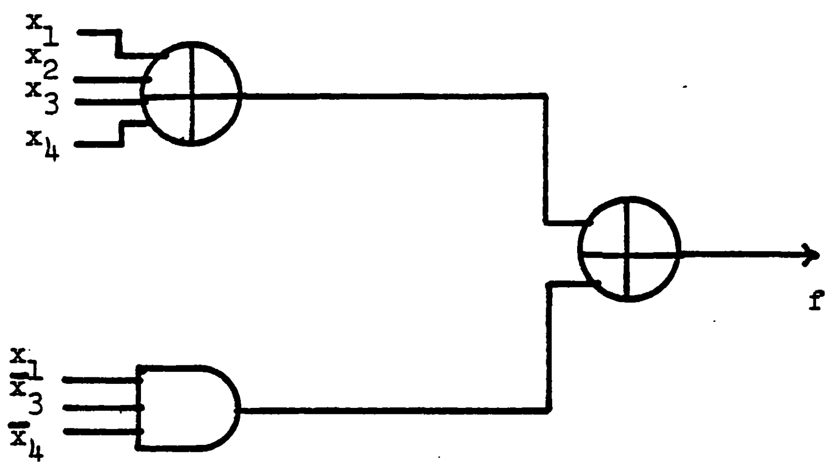


Figure 3.2.e Exclusive-OR decomposition realisation of the example function.

i) The sum of the individual costs of every basic gate in the circuit.We should consider the same type of gates with different number of inputs,as different basic gates (primitives)

ii) Time delay: the maximum number of gates encountered when passing from an arbitrary input to an arbitrary output.

Let us examine,from the cost point of view,the circuits of Figure 3.2 and the corresponding design methods:

a. The 3.2.a synthesis of the function f was determined by minimizing the product-sum expansion ,that is ,determining the minumum number of highest-order cubes (essential prime implicants) to cover the whole function[1,2,3]. Cost: 2 three-input AND; 4 four-input AND; 1 seven-input OR; 4 NOT gates.Delay:2.

b. The 3.2.b circuit was synthesized from the Reed-Muller expansion of a Boolean function [4,5,6]. Once the coeffecients of this expansion are found,then the circuit may very easily be designed. Cost: 4 two-input exclusive-OR gates; 1 two-input AND gate and 1 three-input AND gate.Delay:4.

c. The 3.2.c circuit was synthsized by using the Spectral techniques [7]. Cost: 3 two-input exclusive -OR ; 1 $\langle 2,1,1,1;3 \rangle$ four-input Threshold gate.Delay: 3

d. The 3.2. d circuit was designed by symmetry techniques [8] in the spectrum domain.Cost: 2 two input NAND gates; 2 two-input exclusive-OR gates,and 7 NOT gates.Delay:4.

e. The 3.2.e design is another one with cost of 1 four-input exclusive-OR ; 1 two-input exclusive-OR,and 1 three-input AND, 2 NOT gates.Delay:2.

Since all these circuits(or correspondingly methods) realise the same Boolean function f, to choose the lowest cost (or minumum complexity)circuit we need to define " complexity " of a Boolean function according to different techniques.Then we would be able to select the lowest cost design by calculating this complexity rather than designing the circuits by many different methods.

Another application of the concept of complexity may be the comparison of two different Boolean functions under the same or different design techniques.

So far we have been looking at the need for a concept of complexity from practical point of view.Now let us briefly look through the theoretical developments of this concept in recent decades[9,10]. Firstly ,in 1949 Shannon [11] made an analysis of the complexity for two-terminal switching function realisations. In 1956 Muller [12] further considered the idea of complexity in electronic switching circuits. Muller defined complexity as the sum of the individual costs of every basic gate in the circuit. He proved the relationship below:

$$K_1 \Phi_1 \leqslant \Phi_2 \leqslant K_2 \Phi_1 \qquad \ldots 3.1$$

where Φ_1 is the minumum cost for a Boolean function realisation with certain basic gates and Φ_2 is another minumum cost for the same function realisation with another basic gates set.The constants K_1 and K_2 depend only upon the basic gates and their cost but not the function itself . This relation gives the boundaries for the cost Φ_2 , and also shows that ; to a constant multiplying factor, Φ_2's behaviour is independent of the basic gates which are used.Again,in the same paper a similiar relationship was shown when the number of arguments (p) and the number of Boolean functions (q) are allowed to increase, this being:

$$K_1 \; E_1 (p,q) \leqslant \; E_2(p,q) \leqslant K_2 \, E_1(p,q) \qquad ...3.2$$

where E(complexity) is defined as the maximum cost of the functions for circuits having p inputs and q outputs. K_1 , K_2 constants are the same as described above. Muller further estimated $E_1(p,q) = 2^r/ r$ ($r= p + \log_2 q$) for the boundaries of $E_2(p,q)$ complexity. Shannon [11] and Lupanov [13] also considered complexity boundaries. Later Winograd [14] and Spira [15] both considered "time complexity" (for definition see parameter ii above). In 1975 Davio and Quisquater [16] developed Muller's work for many-valued logic.

We may ask why research on complexity developed in this way,viz. finding boundaries rather than exact complexity values. Infact Yablovskii [17] has conjectured that the labour involved in determining the exact value of complexity is roughly of the same order magnitude as that required to construct a minimal network.

However , while the research on complexity boundaries continued Kellerman [18] considered the average complexity(cost) for a one-output combinational logic network which implements a Boolean function with u "1" vertices, h "0" vertices and n independent variables.He derived an experimental formula for the average cost C:

$$C = K_1 (K_2)^n \frac{u.h}{u + h} = K_1 (K_2)^n \frac{u(2^n-u)}{2^n} \quad ...3.3$$

K_1 and K_2 constants depend on the methods being used.

In 1973 Cook and Flynn [19] suggested an entropy function,which confirms Kellerman's experimental results, as an average cost of a Boolean function. In their paper a single-output binary Boolean function f is treated as a deterministic function of equiprobable inputs and therefore the probability that f=1 (i.e. has output equal to 1) is given by : $P(f) = u/2^n$ where u is the number of "1" vertices and n is the number of arguments.Shannon's [11] entropy function H ,which was proposed as an average cost function of f,was defined in the standard way as

$$H(f) = \frac{u}{2^n} \log_2 \frac{2^n}{u} + \frac{2^n-u}{2^n} \log_2 \frac{2^n}{2^n-u} \quad ...3.4$$

Figure 3.3. shows the maximum ,minumum and average cost(entropy) curves Versus the u "1" vertices (number of true minterms) for three variable ,one output binary Boolean functions.The cost curve of Figure 3.4. is for six-variable functions.

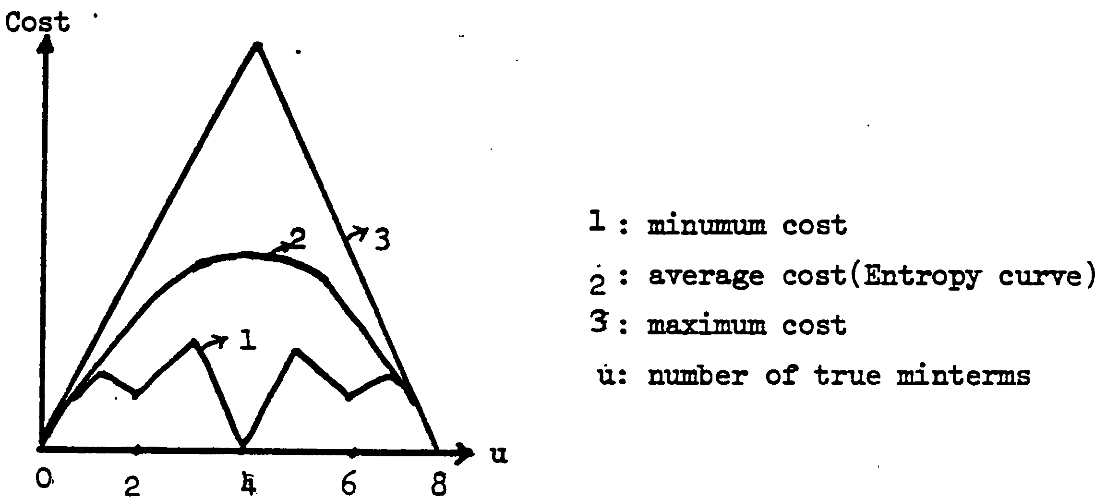


Figure 3.3. Max.,Min. and average cost functions for three-variable Boolean functions.

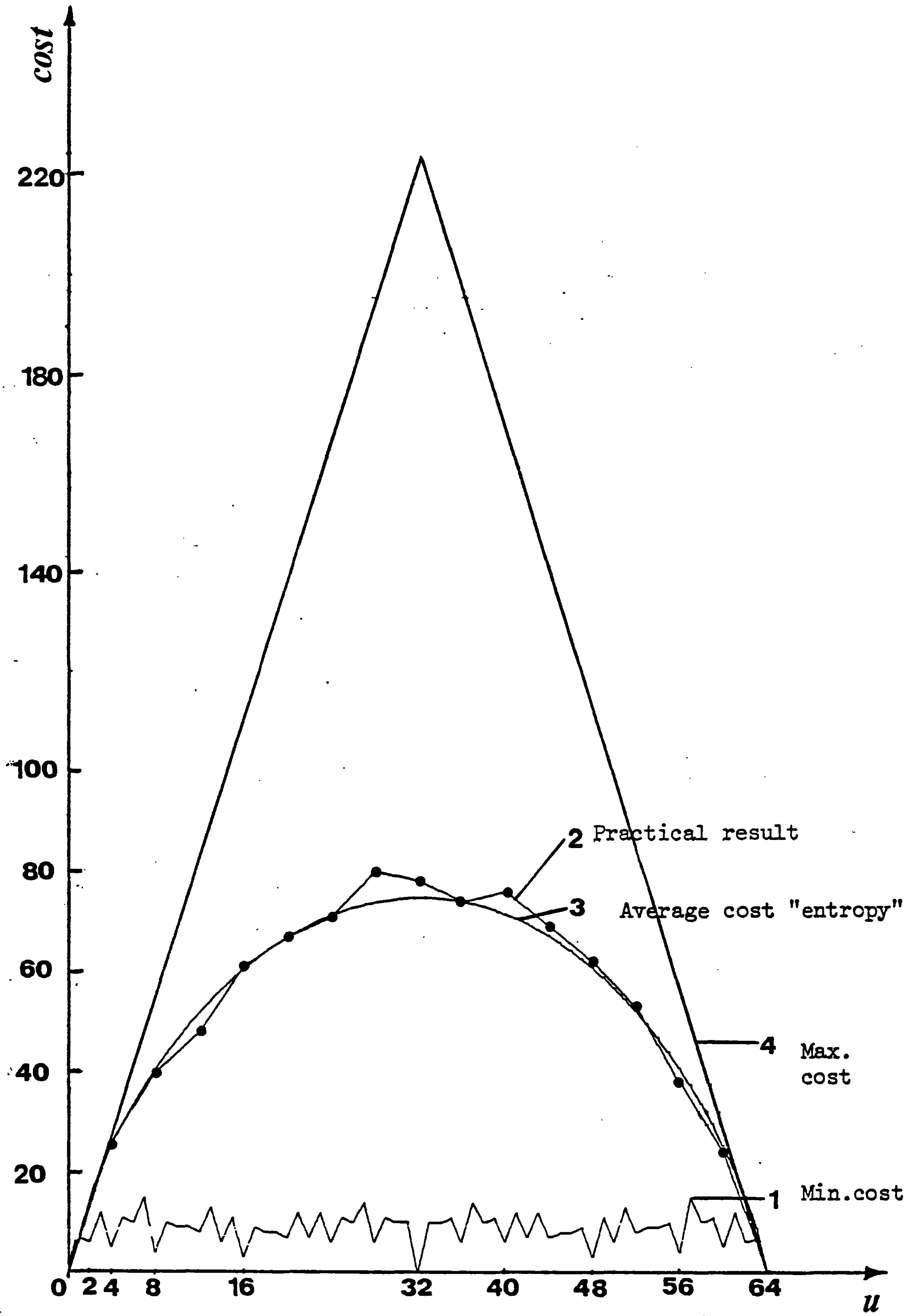


Figure 3.4. Max.,Min.,and average cost-functions versus ' u ' number of true-minterms for six-variable Boolean functions.

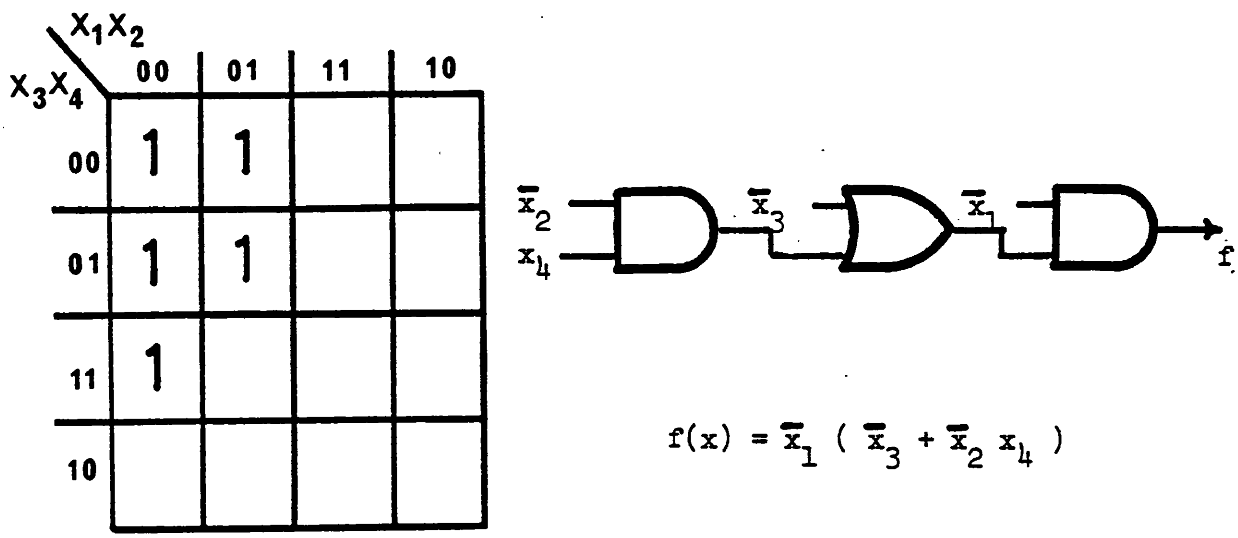
Hellerman [20] ,Mileto and Futzolu [21] and Sholomov [22] have also investigated the probabilistic average cost (or complexity) of binary Boolean functions.

Corollary 3.1. From the above explanations concerning the complexity of a Boolean function it is concluded that, statistically, high or low-number of true (false)minterm functions are less complex than others.

3.1.2. Simplest Functions and Simplest-Threshold Functions

Definition 3.1 For a given number of true minterms (u), there are $\binom{2^n}{u}$ different n-variable Boolean functions which have u true-minterms.Some of these functions can be realised by minumum number of AND or OR gates having two inputs. They are called "simplest functions".

Example 3.1. The function in Figure 3.5 is a simplest function with four variables and five true-minterms(or eleven false-minterms)



$$f(x) = \bar{x}_1 (\bar{x}_3 + \bar{x}_2 x_4)$$

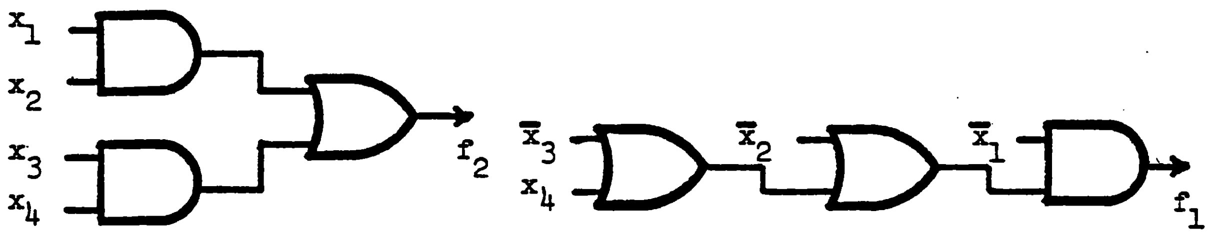
Figure 3.5. A simplest four-variable function with u=5 number of true-minterms.

Henceforth we denote the number of true-minterms by "u" and the number of false-minterms by "v".

Corollary 3.2. N P N (Negation of overall function, Permutation of variables, Negation of variables) manipulation (see reference 23) of a simplest function gives another simplest function.

Corollary 3.3. For a given value of "u", all the simplest functions are not in the same NPN class [23]. That is, there may be two simplest functions with the same number of true-minterms, but one of them cannot be generated from the other one by NPN manipulations.

Example 3.2. f_1 and f_2 functions, in Figure 3.6., are both simplest but not in the same NPN class. Both have u=7.



$f_2 = x_1 x_2 + x_3 x_4$
Simplest but not threshold

$f_1 = \bar{x}_1 (\bar{x}_2 + \bar{x}_3 + x_4)$
Simplest and threshold

x_3x_4 \ x_1x_2	00	01	11	10
00			1	
01			1	
11	1	1	1	1
10			1	

x_3x_4 \ x_1x_2	00	01	11	10
00	1	1		
01	1	1		
11	1	1		
10	1			

Figure 3.6. Two different four-variable simplest functions with u=7.

Appendix -A shows all simplest functions with u true-minterms up to five variables together with the positive-canonical first-order spectral coeffecients and circuit structures.

When the simplest functions spectra and the threshold functions spectra, which are shown in Appendix B, are compared, it can be observed that there is at least one simplest function which is also a threshold function, Such a function will be termed "simplest-threshold".

It has been proved [24] that if a Boolean function can be written

$$f(x_1, x_2, \ldots x_n) = x_i\, g(x_1, x_2, \ldots x_{i-1}, x_{i+1}, \ldots x_n)$$

or as

$$f = x_i\, g$$

where $1 \leqslant i \leqslant n$ and function g does not depend upon the variable x_i, then f is threshold if and only if g is threshold. On the other hand a n-variable simplest function with u true-minterms can be generated from a n-1 variable simplest function, with the same u, in the form

$$f_n = x_n\, f_{n-1} ,$$

where f_n=n-variable simplest function, f_{n-1}=(n-1) variable simplest function. x_n may be replaced by $\bar{x}_n$. In order to include the new variable x_n with f_{n-1}, it is clear that we have to add one two-input AND gate to the circuit realising f_{n-1}. It follows from definition 3.1. that f_n derived from f_{n-1} would be simplest as well. Further if f_{n-1} is Simplest-threshold then f_n will also be Simplest-threshold.

Example 3.3. A fifth-order simplest-threshold function f_5 with u= 5 in Figure 3.7.b is derived from the fouth-order simplest-threshold function f_4 (u=5) in Figure 3.7.a.

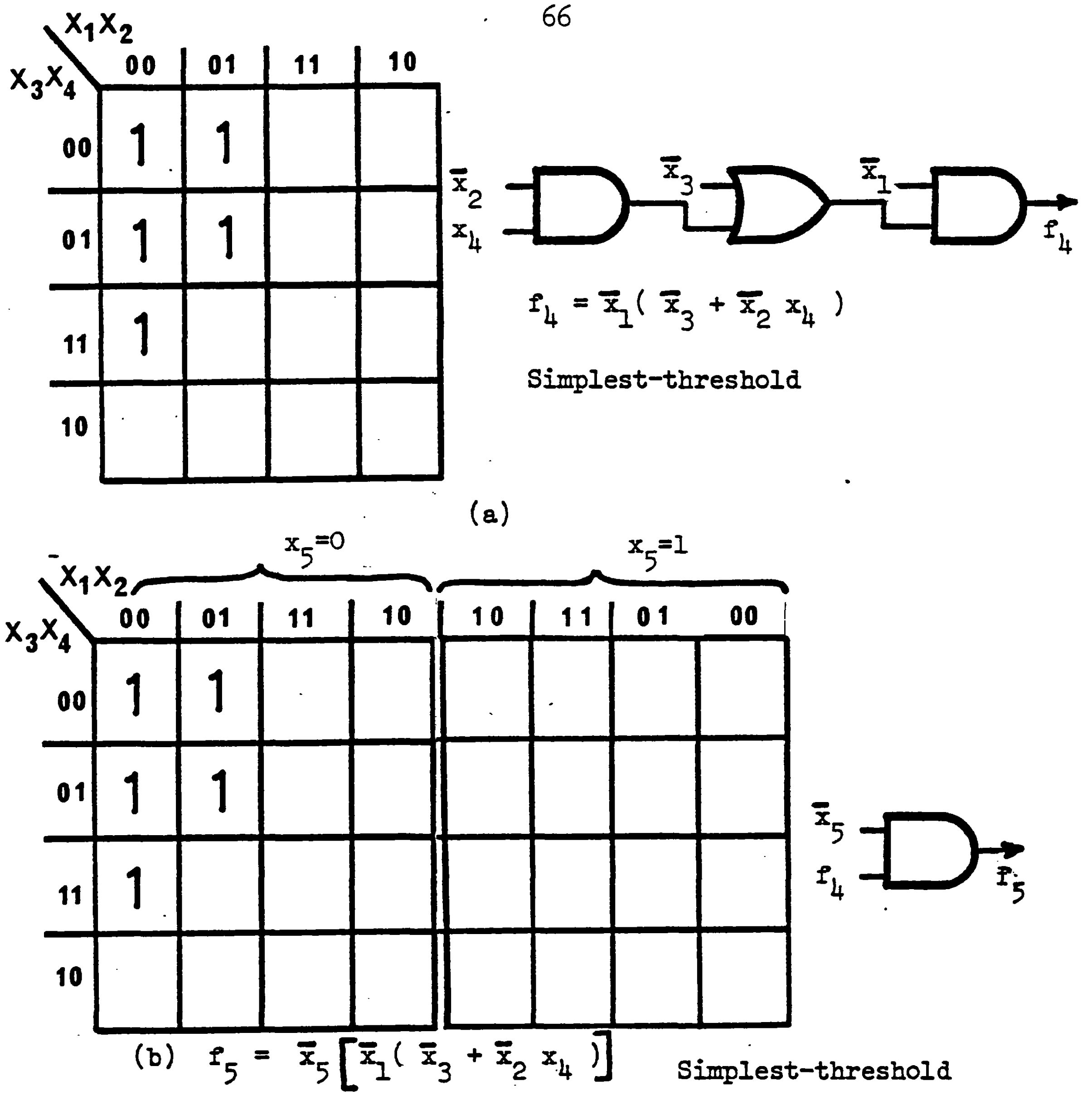


Figure 3.7. Comparison of fourth and fifth order simplest-threshold functions with the same $u=5$.

Corollary 3.4. For a given u ,at least one n-variable simplest function can be found which is threshold,provided we know a lower-order simplest-threshold function with the same u true-minterms.

Corollary 3.5. The first-order spectral coeffecients of a new higher-order n-variable simplest-threshold function (f_n) will be the same as its lower counterpart (f_{n-1}), except for R_0 and R_n; R_n corresponds to the new independent variable x_n,whilst R_0 covers the additional minterms of f(x).These spectral coeffecients can be calculated as

$$R_0 = 2^n - 2u$$
$$R_n = 2u \qquad \ldots 3.5$$

The proof follows from the relationship between the spectral coeffecients and the distribution of minterms, as explained in Chapter 1.

It should be carefully noted that although we have introduced the terminology "simplest-threshold",this does not necessarily require us to use threshold logic gates in a practical realisation.The "simplest threshold " is a target specification for simplest functions and when necessary they can be realised with normal vertex gates,as shown in Appendix A.

Corollary 3.6. Some of the simplest functions are not threshold.

For example, f_2 function in Figure 3.6 is simplest but not threshold. f_1 is simplest-threshold.

3.2. Implementation of the Minterm-Interchange Operation for the Realisation of Boolean Functions

3.2.1. Minterm-Interchange and Design Structure

It is known from Chapter 2 equation 2.9 that

$$f = \gamma \oplus \delta \quad ,\text{when}.$$

f = Boolean function to be designed

δ = Interchanged function,determined by interchanging the true-minterms with false ones in f.

γ = Changer function ,composed of the interchanged true and false minterms.

A possible implementation of equation 2.9 is shown in Figure 3.8 below.

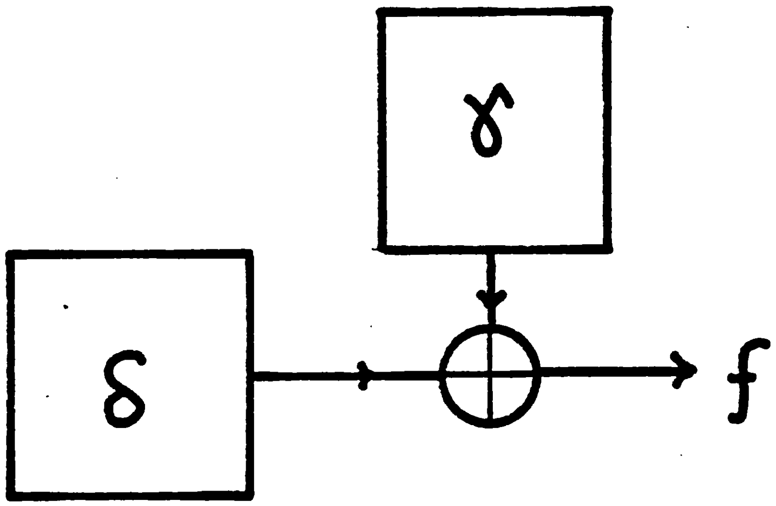


Figure 3.8 Minterm-interchange Design Structure.

In this type of decomposition of f,our aim is to find γ and δ functions with the overall minumum cost by interchanging minterms in f.Let us consider these δ and γ functions seperately.

Firstly, since δ is determined by minterm-interchange in the original function f,the number of true-minterms of both δ and f is the same. Secondly,it would be ideal to choose δ as a simplest-threshold function with the same n and u.Then the cost of δ would be the lowest among the same u true-minterm functions,and δ can easily be recognised by (n+1) first-order spectral coeffecients only. Thirdly,the circuit realisation of the chosen simplest-threshold δ function is universal.That is ,for a given u and n all the particularly chosen simplest-threshold functions realisation circuits are the same.

Now consider the γ function.According to entropy cost mentioned in Chapter 3.1.1.,statistically the cost of this γ function will be low if it has small (or high) number of true-minterms.

Hence in the light of the above explanations we can state the general rule for the design of combinational logic circuits using minterm-interchange operation.

Corollary 3.7. The minumum possible number of minterms of f should be interchanged (resulting in the lowest cost of γ) in such a way that we are able to generate δ as simplest-threshold function.

This is illustrated on the entropy cost curve in Figure 3.9.a. C_γ , C_δ , C_f are the costs of γ, δ and f functions respectively. The total cost of this decomposition will be:

$$C_{total} = C_\gamma + C_\delta + (\text{ cost of two-input exclusive-OR}) \quad \ldots 3.6$$

As a result it is normally expected that:

$$C_{total} \leqslant C_f$$

for the minterm-interchange design technique. In general, from the entropy cost curve, we can see that this decomposition would be useful when f is close to the centre of the u-axis, and has an above average cost. Figure 3.9.a,b,c show different total cost examples for differently distributed functions on the entropy curve.

It is important to note that we always use the left-hand side of entropy cost curve, since if f is on the right-hand side ($u < v$) then we can consider the complementary function $\bar{f}$.

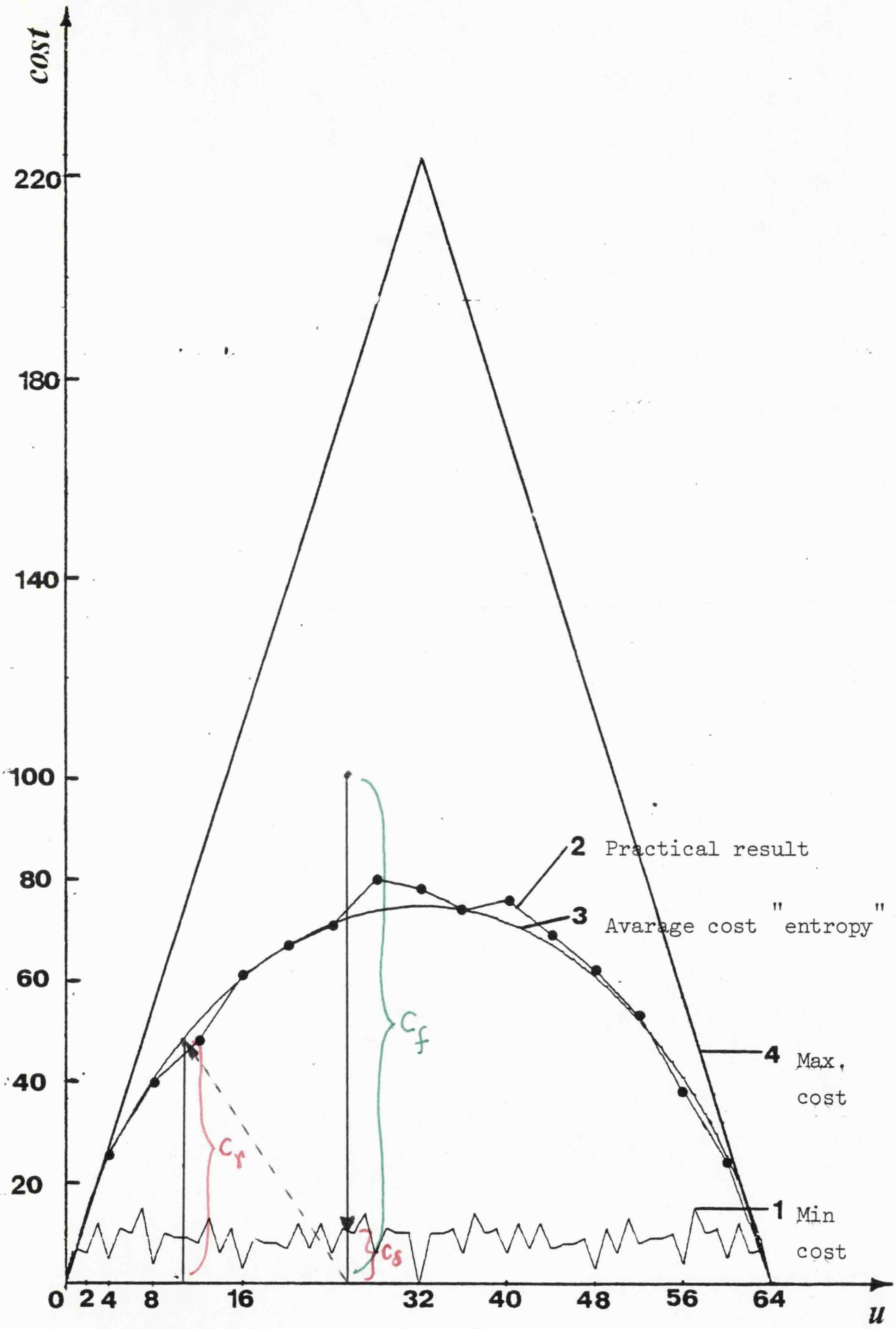


Figure 3.9.a The best situation to apply minterm-interchange design technique.f is close to the centre of u-axis and has above average cost.

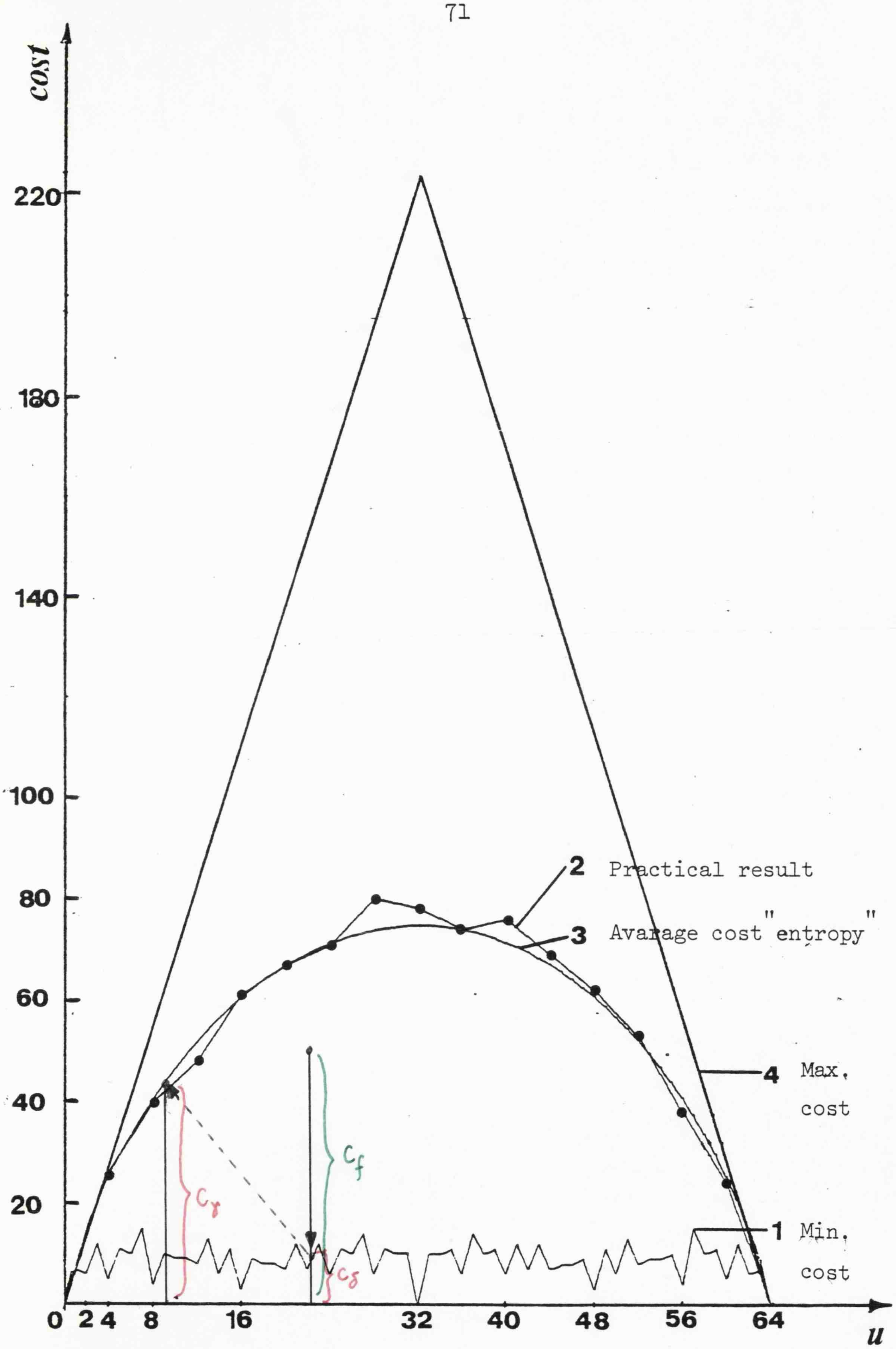


Figure 3.9.b f is close to the centre of u-axis but has a lower cost than average.Critical situation to make a decision on applying minterm-interchange.

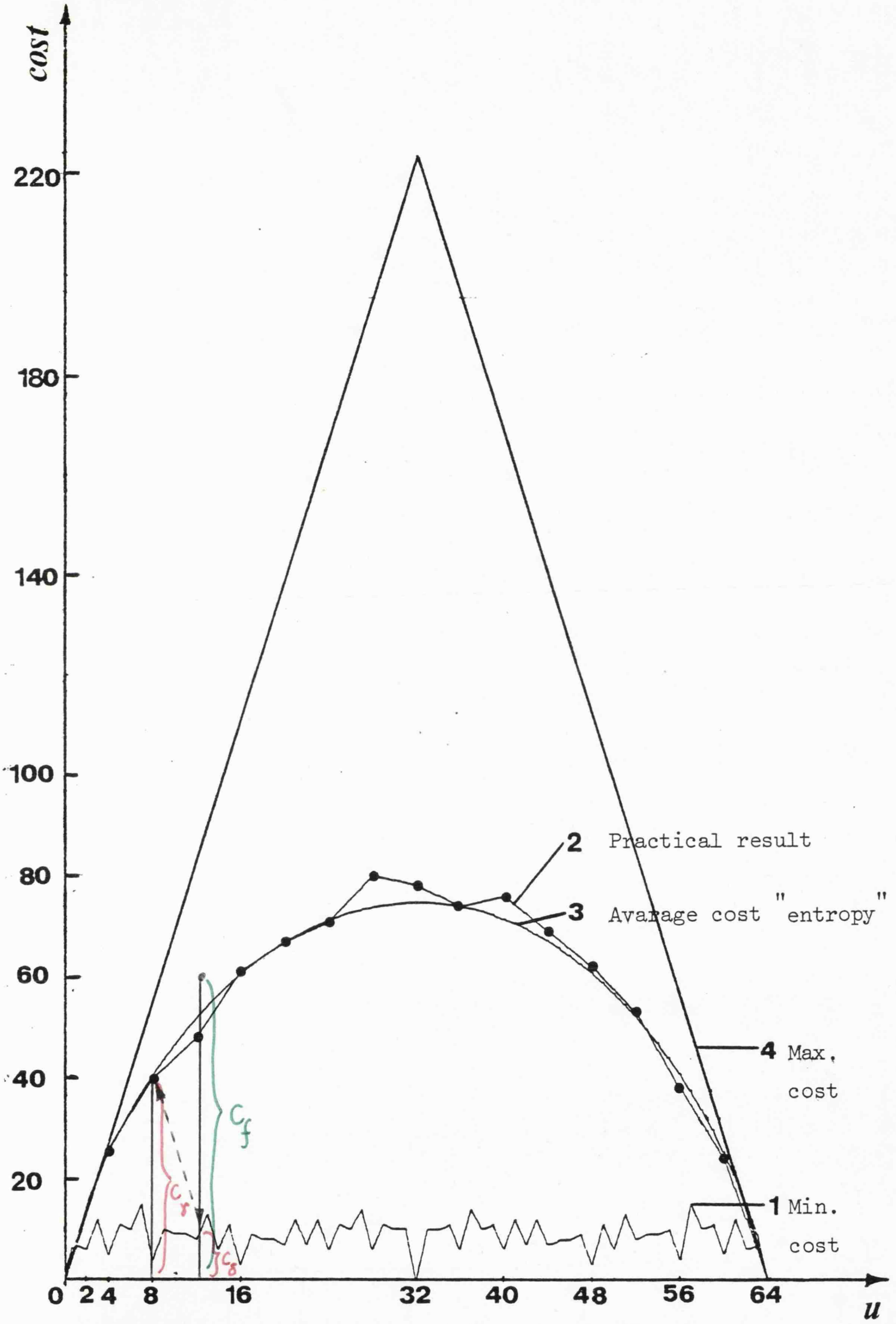


Figure 3.9. c f is not close to the center of u-axis but has an above average cost. Critical situation to make a decision on applying minterm-interchange.

3.2.2. Minterm-interchange Decomposition in the Spectrum Domain

In the minterm-interchange decomposition procedure,which will be described in next section,simplest functions(δ)can be chosen as either threshold or as Chow-unique functions[22].In our case we shall only use threshold functions. The reasons are firstly that it has already been shown there is at least one simplest-threshold function for a given number of u-true-minterms (corollary 3.4),and secondly it is easy to manipulate threshold and Chow-unique functions in the spectrum domain.This is because these functions are uniquely defined by their primary group of (n + 1) first-order spectral coeffecients, which correspond to Chow parameters [23,25] .Now we can modify our design structure to the form below in Figure 3.10.

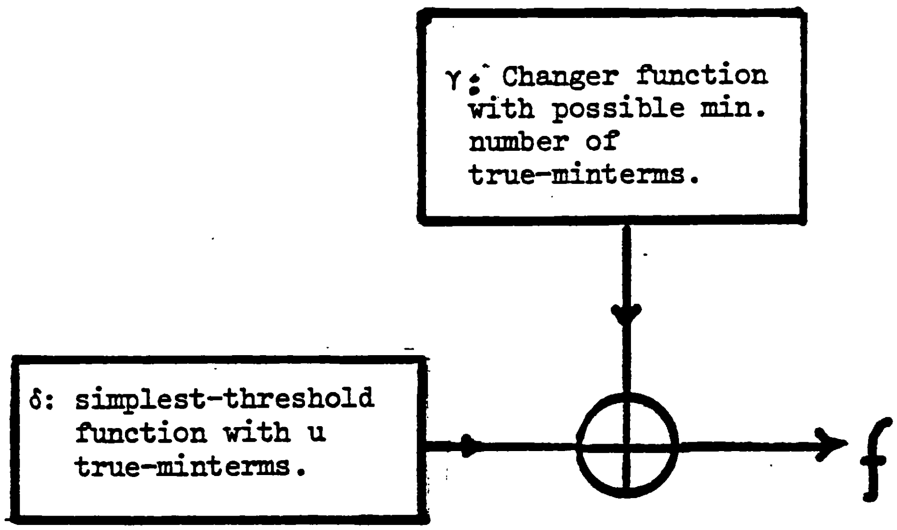


Figure 3.10 Minterm-interchange decomposition with simplest-threshold function (δ) and a low cost function (γ).

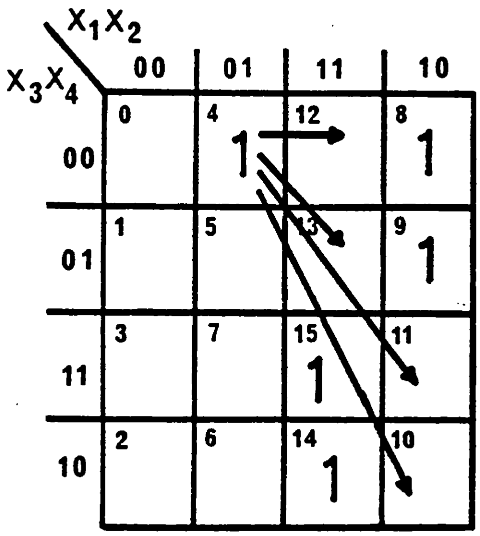
Since interchanging a pair of minterms (one true, one false) can effect the first-order spectral coeffecients of the function f by ± 4 (corollary 2.3),we can formulate a minumum possible number of minterm-pair interchanges (which we shall denote by k/2 henceforth) to find the simplest-threshold δ function.Note that:

$\frac{k}{2}$ = minumum possible number of minterm-pair interchanges,

= $\frac{1}{4}$ { [absolute value of the highest first-order spectral coeffecient of the simplest function δ] - [absolute value of the highest first-order spectral coeffecient of f] } ...3.7

If the expression in the brace is zero then we can choose the second highest spectral coeffecients.

Let us consider,in more detail,what is meant by "minumum possible number of minterm-pair interchanges". The k/2 value obtained from equation 3.7 may not be adequate to generate a simplest-threshold function δ .For example let us take the function f in Figure 3.11.



first-order spectral coeffecients of f:

(6 2 2 2)

first-order spectral coeffecients of δ:

(10 6 2 2)

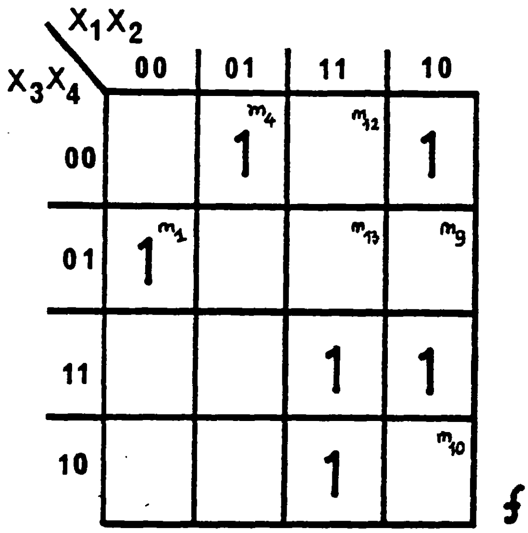
Figure 3.11 An example for (k/2)=1 minumum possible number of minterm-pair ,but inadequate to generate the simplest-threshold function δ .

We find (k/2)=1 by equation3.7,that is we need to change at least one minterm-pair to obtain the simplest-threshold function.This pair may be the interchange of true-minterm m_4 with one of the m_{10},m_{11},m_{12} m_{13} false minterms.However none of the replacements gives a simplest-threshold function.In this case we shall need to add another step to the design procedure,which will be explained in the next section.That is why we defined k/2 as the minumum possible number of minterm-pair interchange.

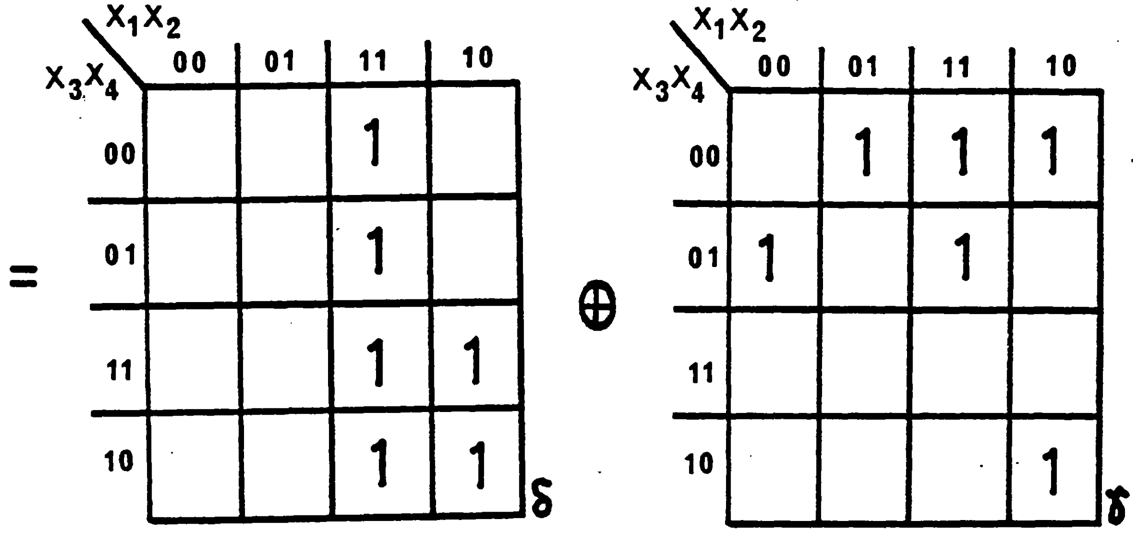
3.2.3. Design Procedure

For a given (u) number of true-minterms, the simplest-threshold functions are invariant under N P (negation,permutation) manipulations(corollary 3.2).So we are confronted with the problem of choosing the simplest-threshold δ among a NP simplest-threshold class of functions which gives the function Υ with k minimum possible number of true minterm when exclusive-OR 'ed by f(x),see Figure 3.I0.In order to determine the δ function which obeys the rule mentioned in corollary 3.7.,a comparison between the first-order coeffecients of δ and f , and changing the signs and the positions (NP manipulation)of the first-order spectral coefficients of δ to coincide with the first-order spectral coefficients of f could be suggested.But a counter example will be given which proves that it is not possible to predict whether the number of true-minterms of Υ will be k minumum possible or not,unless the exclusive-or operation between all possible members of NP class of δ and f are executed.

Example 3.4 Let us take the function f in Figure 3.12.a which has six true-minterms (u=6). The first-order spectral coefficients of f are (4 0 0 0) and those of the simplest-threshold function δ $[\delta = x_1(x_2 + x_3)$ with u=6 $]$ are(12 4 4 0) written in the order R_1,R_2,R_3,R_4 successively.The minumum possible number of



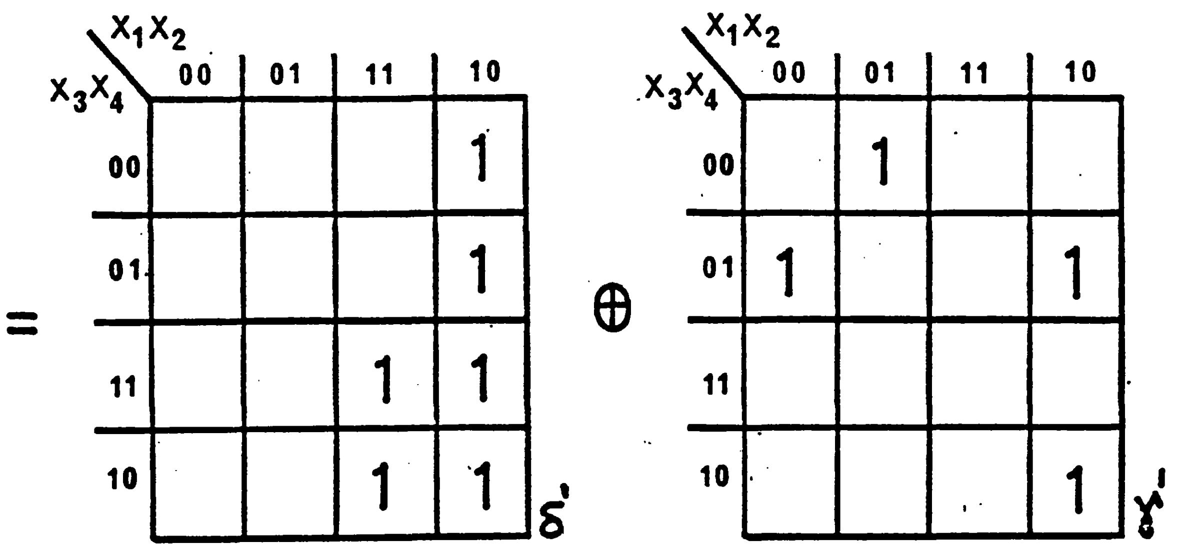
(a) given function f



$\delta = x_1(x_2 + x_3)$ $\qquad u_\gamma = 6$

(12 4 4 0)

(b) δ chosen by comparison with f, and the related γ function



$\delta = x_1(\bar{x}_2 + x_3)$ $\quad u_{\gamma'} = 4$

(12 -4 4 0)

(c) δ generated by Negation manipulation on δ,and related γ function

Figure 3.12 A counter example to show that NP manipulation on the first-order spectral coeffecients of simplest-threshold function δ to coincide with the first-order spectral coeffecients of f would not necessarly give the γ function with minumum possible number of true-minterms.

minterm-interchange pairs k/2 is equal to $\frac{12-4}{4} = 2$ (equation 3.7). That is γ should have at least four true-minterms. After comparing the δ and f spectra , it is possible to choose δ as in Figure 3.12.b.This δ gives γ with $u_\gamma = 6$ true-minterms which is more than k = 4 minumum possible true-minterms.However the δ' ,determined by $x_2 \leftrightarrow \bar{x}_2$ Negation operation on δ ,gives γ' with $u_{\gamma'} = 4$ true-minterms. That is γ' has the minumum possible number of true-minterms k = 4.

Clearly we have found different numbers of true-minterms (u_γ , $u_{\gamma'}$) in the γ functions (i.e. compare γ , γ')when we perform N manipulation on the δ function.Because of this diffuculty of predicting the δ which gives k minumum possible number of true-minterms in γ , an alternative approach employing equation 2.23 will be explored.

Let us follow this prefered minterm-interchange decomposition procedure step by step:

The notations which are to be used in this procedure are:

f: Boolean function to be designed

n: number of variables

u: number of true-minterms of f.

δ^*: Simplest-threshold function, with u true-minterms inthe form given in Appendix A.

δ: Simplest-threshold function which is obtained from δ^* by NP manipulations. δ is to be used in the final circuit.

$\underset{\sim}{S}_{\delta}^{(c)}$: Canonical first-order spectral coeffecients of δ which has the same form given in Appendix A

$$\underset{\sim}{S}_{\delta}^{(c)} = \left[R_1' \quad R_2' \quad \dots . \ R_i' \quad \dots R_n' \right]^t$$

$\underset{\sim}{S}_f$: first-order spectral coeffecients of f

$$S_f = \left[R_1^f , R_2^f \dots . R_i^f \dots . R_n^f \right]^t$$

$S_f^{(c)}$: Canonical form of S_f

$$S_f^{(c)} = \left[R_1 , R_2 \dots . R_i \dots . R_n \right]^t$$

k : minumum possible number of true-minterms in γ

ℓ : number of possible false-minterms to be interchanged by k number of true-minterms in f.

Step 1

If $u > 2^{n-1}$ then we shall design $\bar{f}$, that is we are on the left-half-side of entropy cost curve. For $u < 2^{n-1}$ design f.

Step 2

Find the first-order canonical spectral coeffecients of the Boolean function f. Calculate

$$\frac{k}{2} = \frac{R_1' - R_1}{4} \qquad \dots 3.8$$

If $k/2 = 0$ then we compare second highest spectral coeffecients of δ and f to decide the minumum possible number of minterm-interchange pairs. Say i th canonical spectral coeffecient of f gives k/2 value, which is non-zero. R_i^f correspond to R_i in canonical form of $\underset{\sim}{S}_f^{(c)}$. There are two cases:

2.a) If $R_i^f < 0$

increase the R_i^f value up to the R_i value of δ, by interchanging (k/2) true-minterms in $x_i = 1$ space* with false-minterms in $x_1 = 0$ space. These true-minterms can be found by examining the true-minterms which have " 1 " in i th bit positions.

2.b) If $R_i^f > 0$

Replace the $x_i = 0$ space by the $x_i = 1$ space and vice versa, and " 1 " in the i th bit positions by " 0 " in the i th bit positions.

In the case $R_i^f = 0$ either of the two methods associated with cases a,b above may be employed.

Definition 3.2 True-minterm column matrix : sum of the column matrices correspond to M_i true-minterms in <1,-1> domain which are to be interchanged (see equation 2.23), i.e.

$$\underset{\sim}{x} = \sum_{i=1}^{k/2} \underset{\sim}{M}_i$$

Order of $\underset{\sim}{x}$ is n×1 .

Steps 1 and 2 for the function in Figure 3.12 are as follows

$u=6 < 2^{4-1} = 8$

$$\frac{k}{2} = \frac{R_1' - R_1}{4} = \frac{12-4}{4} = 2$$

$R_1^f = 4 > 0$

The two true-minterms in x_1=0 space are m_1(0 0 0 1) and m_4(0 1 0 0)

---

* see Appndix D for the spaces for n=4

$$\underset{\sim}{x} = \overset{M_1}{\begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}} + \overset{M_4}{\begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}} = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

Step 3

3.a) If $R_i^f < 0$

Find the possible false-minterms to be interchanged in $x_i = 0$ space. The number of these possible false-minterms is

$$\ell = 2^{n-1} - \left(u - \frac{k}{2} \right) \qquad \ldots 3.9$$

We can determine these false-minterms by searching for the false-minterms which have " 0 " in i th bit positions.

3.b) If $R_i^f > 0$

Replace $x_i = 0$ space by $x_i = 1$ space and "0" in the i th position by "1" in i th positions.

Definition 3.3 False-minterm matrix Q : Composed of the columns which correspond to the ℓ number of possible false-minterms to be interchanged.This matrix is in the $\langle 1,-1 \rangle$ domain with $n \times \ell$ order.

For the previous example:

$R_i = 4 > 0$

$\ell = 2^{4-1} - (6-2) = 8-4 = 4$

ℓ number of possible false-minterms in $x_1 = 1$ space are $m_9 = (1\ 0\ 0\ 1)$ $m_{10} = (1\ 0\ 0\ 1)$, $m_{12} = (1\ 1\ 0\ 0)$, $m_{13} = (1\ 1\ 0\ 1)$.

Now,we can write the false-minterm matrix Q

$$Q = \begin{array}{cccc} M_9 & M_{10} & M_{12} & M_{13} \\ \end{array}$$

$$Q = \left[\begin{array}{cccc} -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{array}\right]$$

Step 4

Definition 3.4 Combination matrix C_d : $\ell \times 1$ order column matrix which has $k/2$ number of "1" entries and $(\ell - \frac{k}{2})$ number of "0" entries.

Number of such matrices is

$$d = \binom{\ell}{k/2} = \frac{\ell!}{(\ell - \frac{k}{2})!\,(\frac{k}{2})!} \qquad \ldots 3.10$$

For example $\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^t$ and $\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^t$ are two combination matrices with $\ell = 4$, $(k/2) = 2$.

Using equation 2.23

$$\underset{\sim}{S}_d = \underset{\sim}{S}_f + 2 \left(\sum_{i=1}^{k/2} \underset{\sim}{M}_i - Q\, \underset{\sim}{C}_d \right)$$

$$= \underbrace{\underset{\sim}{S}_f + 2\, \underset{\sim}{X}}_{\underset{\sim}{Y}} - 2\, Q\, \underset{\sim}{C}_d$$

$$= \underset{\sim}{Y} - 2\, Q\, \underset{\sim}{C}_d \qquad \ldots 3.11$$

Now evaluating the equation 3.11 for every $\underset{\sim}{C}_d$ matrix until we find the same canonical spectral coeffecients ($\underset{\sim}{S}_d^{(c)}$) with the simplest-threshold functionδ ,we would be able to decide the exact true and false minterms to be interchanged in order to confirm the rule stated in corollary3.7. The positions of "1" entries in this particular $\underset{\sim}{C}_d$ matrix defines the columns of false-minterms that we choose, i. e. γ is determined by step 4.

Step 5

Since the canonical spectral coeffecients of both δ and f are equal.,NP manipulations on simplest-threshold function δ^* can be employed in such a way that spectrum of δ becomes exactly equal to $\underset{\sim}{S}_f$, including the signs and the positions of spectral coeffecients . So the δ ,satisfying our purpose,is defined.

Example 3.5 Design the function f given in Figure 3.13.

X_4X_5 \ X_2X_3	00	01	11	10	10	11	01	00
00	1	1	1	m_8				
01	1	m_5	1	1			1 m_{21}	
11	m_3	1	1	1				1 m_{19}
10	1	1	1	m_{10}				
	$\bar{X}_1$				X_1			

Figure 3.13 An example for minterm-interchange design

Step 1

u= 14 n=5

since $14 < 2^{n-1} = 2^4 = 16$,we will design f.

Step 2

Spectrum of f : $\underset{\sim}{S}_f = \begin{array}{ccccc} R_1^f & R_2^f & R_3^f & R_4^f & R_5^f \end{array} \left[\begin{array}{ccccc} -20 & -4 & 4 & 0 & 4 \end{array}\right]$

Canonical form of S_f : $\underset{\sim}{S}_f^{(c)} = \begin{array}{ccccc} R_1 & R_2 & R_3 & R_4 & R_5 \end{array} \left[\begin{array}{ccccc} 20 & 4 & 4 & 4 & 0 \end{array}\right]$

Spectrum of simplest-threshold δ : $\underset{\sim}{S}_\delta^{(c)} = \begin{array}{ccccc} R_1' & R_2' & R_3' & R_4' & R_5' \end{array} \left[\begin{array}{ccccc} 28 & 4 & 4 & 4 & 0 \end{array}\right]$

$$\frac{k}{2} = \frac{R_1' - R_1}{4} = \frac{28 - 20}{4} = 2$$

Corresponding spectral value $R_1^f = -20 < 0$

$(k/2) = 2$ true-minterms in $x_1 = 1$ space are

$$m_{19} = [1\ 0\ 0\ 1\ 1] \quad \text{and} \quad m_{21} = [1\ 0\ 1\ 0\ 1]$$

True-minterm matrix X :

$$\begin{array}{ccccccc} & & & M_{19} & & M_{21} & \\ \underset{\sim}{X} = \sum_{i=1}^{k/2} M_i & = & & \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} & + & \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} & = \begin{bmatrix} -2 \\ 2 \\ 0 \\ 0 \\ -2 \end{bmatrix} \end{array}$$

Step 3

$$R_1^f = -20 < 0$$

$$\ell = 2^{n-1} - (u - \frac{k}{2}) = 2^{5-1} - (14 - 2) = 16 - 12 = 4$$

$\ell = 4$ number of possible false-minterms in $x_1 = 0$ space are

$m_3 = (0\ 0\ 0\ 1\ 1)$, $m_5 = (0\ 0\ 1\ 0\ 1)$, $m_8 = (0\ 1\ 0\ 0\ 0)$

$m_{10} = (0\ 1\ 0\ 1\ 0)$

False-minterm matrix Q:

$$\begin{array}{cc} & \begin{array}{cccc} M_3 & M_5 & M_8 & M_{10} \end{array} \\ Q = & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \end{array}$$

Step 4

$$d = \binom{\ell}{k/2} = \binom{4}{2} = \frac{4}{2!\,2!} = \frac{3.4}{2} = 6$$

$$\underset{\sim}{Y} = \underset{\sim}{S}_f + 2\underset{\sim}{X} = \begin{bmatrix} -20 \\ -4 \\ 4 \\ 0 \\ 4 \end{bmatrix} + 2 \begin{bmatrix} -2 \\ 2 \\ 0 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} -24 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

$$\underset{\sim}{S}_1 = \underset{\sim}{Y} - 2Q\,\underset{\sim}{C}_1 = \begin{bmatrix} -24 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix} - 2 \begin{array}{c} \begin{array}{cccc} M_3 & M_5 & M_8 & M_{10} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} C_1 \\ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{array} \begin{array}{l} m_3 \\ m_5 \\ m_8 \\ m_{10} \end{array} = \begin{bmatrix} -24 \\ -4 \\ 4 \\ 0 \\ 4 \end{bmatrix}$$

Repeating this calculation for the other five $\underset{\sim}{C}_d$,we obtain different $\underset{\sim}{S}_d$ as follows:

$$\underset{\sim}{C}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^t \qquad \underset{\sim}{S}_2 = \begin{bmatrix} -28 & 0 & 0 & 0 & 0 \end{bmatrix}^t$$
$$\underset{\sim}{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^t \qquad \underset{\sim}{S}_3 = \begin{bmatrix} -28 & 0 & 0 & 4 & 0 \end{bmatrix}^t$$
$$\underset{\sim}{C}_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^t \qquad \underset{\sim}{S}_4 = \begin{bmatrix} -28 & 0 & 4 & 4 & 0 \end{bmatrix}^t$$
$$\underset{\sim}{C}_5 = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^t \qquad \underset{\sim}{S}_5 = \begin{bmatrix} -28 & 0 & 4 & 0 & 0 \end{bmatrix}^t$$
$$\underset{\sim}{C}_6 = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^t \qquad \underset{\sim}{S}_6 = \begin{bmatrix} -28 & 4 & 0 & 0 & -4 \end{bmatrix}^t$$

Examining the $\underset{\sim}{S}_d$ spectra ,we find the $\underset{\sim}{S}_1 = \underset{\sim}{S}_\delta$ corresponding to $\underset{\sim}{C}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$. Then the false-minterms to be interchanged are m_3 and m_5 . We already found the true-minterms to be interchanged, m_{19}, m_{21}, at step 2. The other $\underset{\sim}{C}_2, \underset{\sim}{C}_3, \underset{\sim}{C}_4, \underset{\sim}{C}_5, \underset{\sim}{C}_6$ do not give any $\underset{\sim}{S}_d$ which is canonically equal to $\underset{\sim}{S}_\delta$. Therefore the unique γ function for this example is as in Figure 3.14.

Step 5

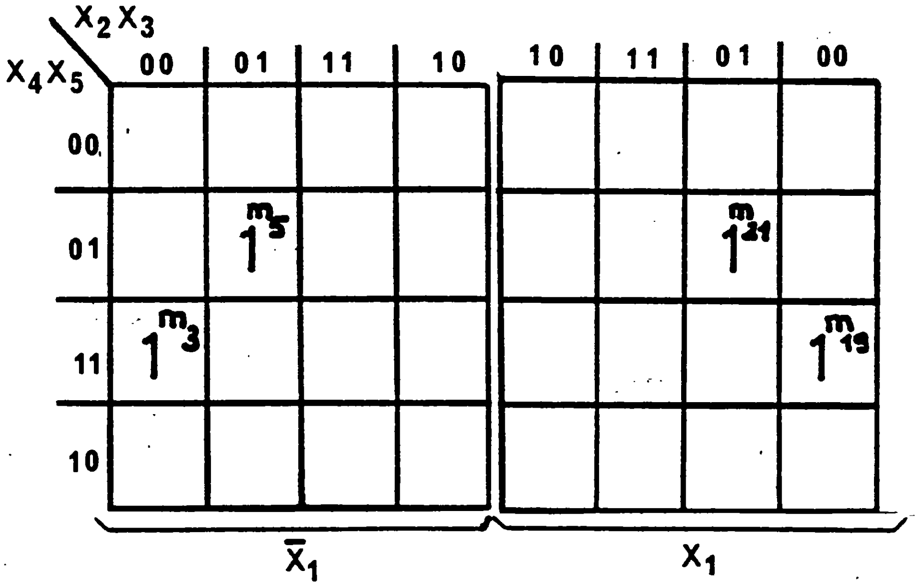
$$\underset{\sim}{S}_1 = \begin{bmatrix} -28 & -4 & 4 & 0 & 4 \end{bmatrix}^t$$
$$\underset{\sim}{S}_\delta^* = \begin{bmatrix} 28 & 4 & 4 & 4 & 0 \end{bmatrix}^t \qquad \delta^* = x_1(x_2 + x_3 + x_4)$$

NP manipulation : $x_1 \leftrightarrow \bar{x}_1$; $x_2 \leftrightarrow \bar{x}_2$; $x_4 \leftrightarrow x_5$

then

$$\underset{\sim}{S}_\delta = \begin{bmatrix} -28 & -4 & 4 & 0 & 4 \end{bmatrix}^t \quad \delta = \bar{x}_1(\bar{x}_2 + x_3 + x_5)$$

Hence δ is determined. The realisation is in Figure 3.15 below.



$$\gamma = \bar{x}_2\, x_5\, (\, x_3\, \bar{x}_4 + \bar{x}_3\, x_4\,)$$

Figure 3.14 Minumum number of true-minterm γ function to generate the simplest function δ out of f by minterm-interchange.

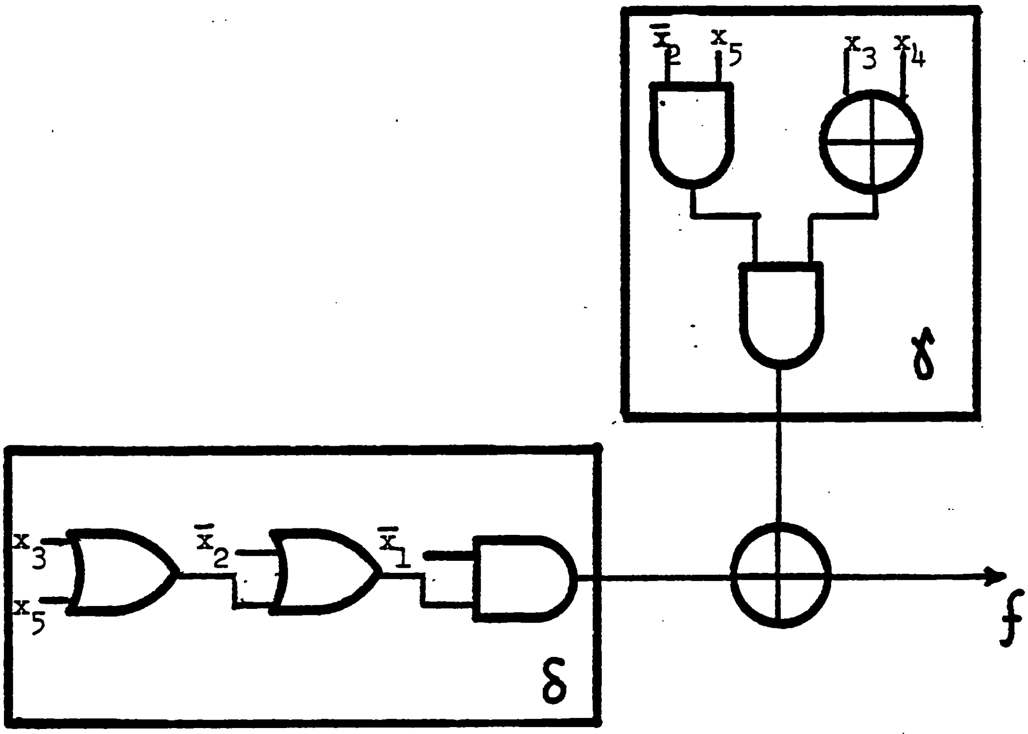


Figure 3.15 Realisation of the f by using minterm-interchange technique.

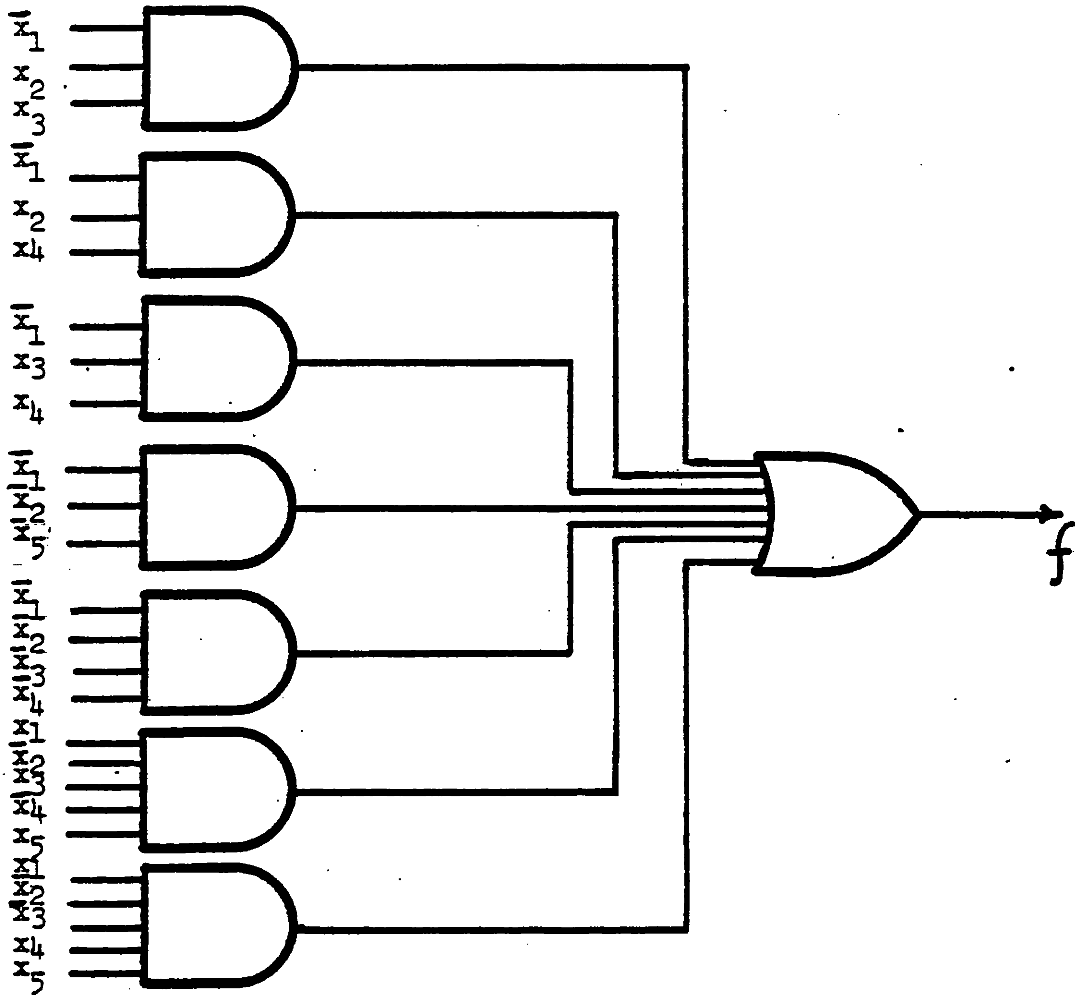


Figure 3.16 Conventional two-level realisation of the function f by using Quine-McClusky **technique.**

We can now compare this minterm-interchange realisation with the conventional two-level realisation shown in Figure 3.16 above:the former realisation cost: 3 two-input AND gates,2 two-input OR gates and 2 two-input exclusive Or gates,delay:4 ; The latter,conventional, realisation cost:4 three-input AND gates,1 four input AND gate,2 five-input AND gates and 1 seven input OR gate.Delay 2.The NOT gates are not taken into account,but would be three for Figure 3.15 realisation,and fourteen for Figure 3.16 realisation.

3.3 Combining Minterm-interchange and Spectral Translation Methods

3.3.1 Comparison of Minterm-interchange and Spectral Translation

In 1975 it was shown [7] that,using spectral translation, it is possible to generate a new function from the spectrum of a given function in such a way that both spectra have the same spectral coeffecient values but in different positions and/or signs (see Chapter 1.4) This spectral translation has also been implemented for Boolean function realisation by exclusive-OR gates in the form in Figure 3.17.

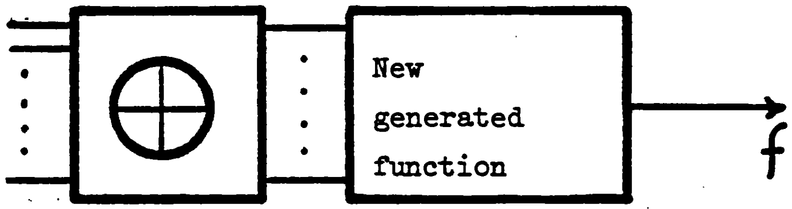


Figure 3.17 Spectral translation circuit structure

Some of the properties of spectral translation are [7, 23]

i) The R_0 ,which determines the number of true-minterms(or false) in the function,is not changed by this translation.

ii) Translation of the adjacent-order coeffecients correspond to changing half of the minterm positions (true or false or both). Only certain minterm-interchanges are allowed under this translation. In Appendix E , all second order spectral translations are given for four-variable functions.

iii) When the spectral translation technique is used for design purposes, if two adjacent-order spectral coeffecients are interchanged then the circuit structure will be as in Figure 3.17 save that the exclusive-OR gate will have only two inputs.

iv) A new function ,with spectral coeffecient values which do not exist in the original function spectrum,cannot be generated by spectral translation techniques, i.e. the spectra of both functions will be the same excepting the positions and signs

Let us compare the properties of minterm-interchange below with those of spectral translation above.

i) R_o does not change ,i.e. number of true-minterms u is the same under this operation.

ii) Interchanged minterms are not limited.That is any two minterms can be changed.The interchange of a true and false minterm effects half of the spectral coeffecients.

iii) Interchanging a pair of Hamming-distance one minterms (one false,one true) results in a n-1 input AND gate,two-input exclusive OR gate,and the new function when a minterm-interchange design is considered. In the worst case the n-1 input AND gate is replaced by two n-input AND gates.

iv) It is possible to create spectral coeffecients which do not exist in the original function spectrum.

3.3.2 Procedures for Combined methods

Spectral translation is more useful then minterm-interchange when the function is simplest embedded-threshold [*,7] because in this case it is possible to obtain the simplest-threshold function by interchanging more than one minterm pair at a time by employing exclusive-OR gates.However the simplest-threshold function cannot be generated by spectral translation when the original function spectrum does not contain

* For definition and further details see Chapter 1.4.

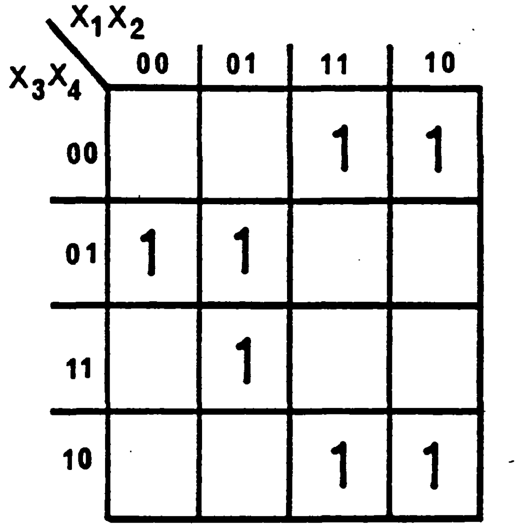
all the first-order spectral coeffecients of δ .Therefore in these cases minterm-interchange operation is essential.

Example 3.6 A simplest-embedded-threshold function is given in Figure 3.18.a By $R_{14} \leftrightarrow R_1$ spectral translation we can realise the function f as in Figure 3.18.b.If we used the minterm-interchange technique,the circuit would be as in Figure 3.18.c.Thusfor this particular function realisation spectral translation is preferable.

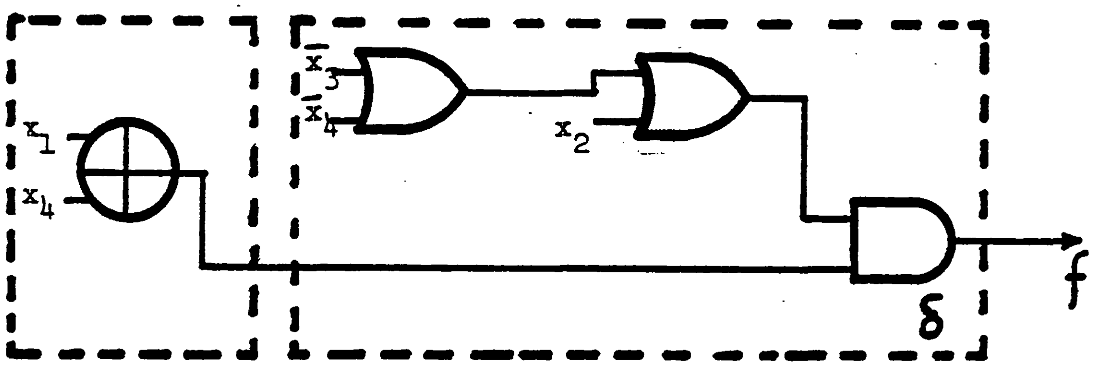
spectrum of f:

R_0	R_1	R_2	R_3	R_4	R_{12}	R_{13}	R_{14}
2	2	-2	-2	2	2	2	14

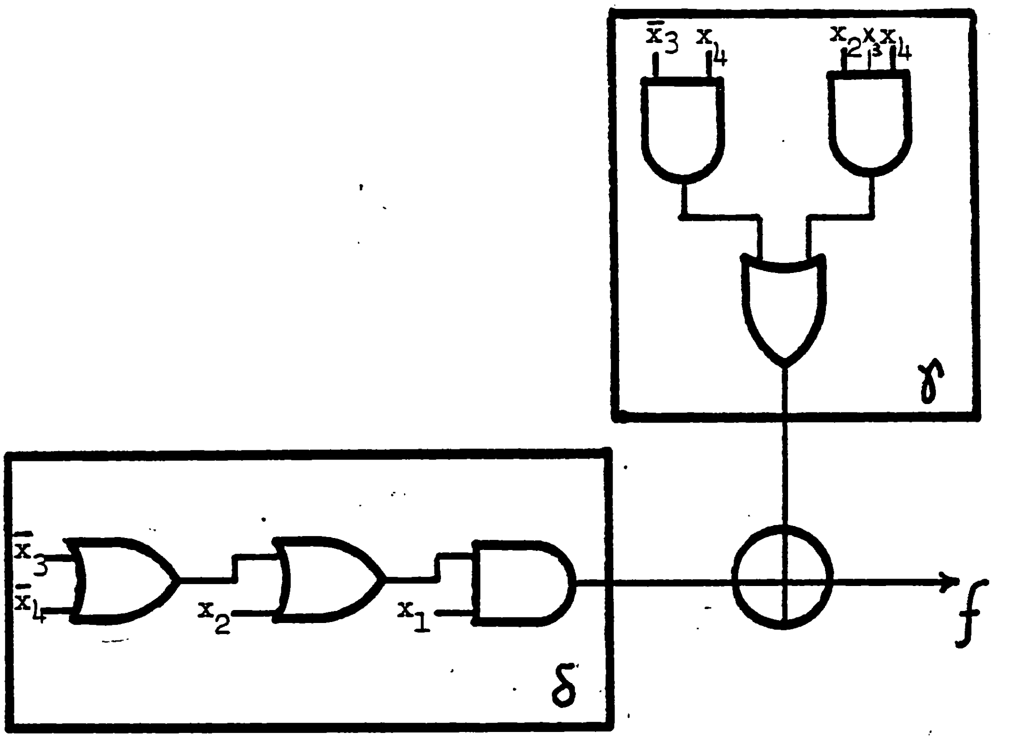
R_{23}	R_{24}	R_{34}	R_{123}	R_{124}	R_{134}	R_{234}	R_{1234}
-2	-2	-2	-2	-2	2	2	2



(a) given function f



(b) Spectral translation design of the function f.



(c) Minterm-interchange design of the function f.

Figure 3.18. Two different realisation; one spectral translation (b) and the other minterm-interchange(c) for the same function f.

Example 3.7 For the function given in Figure 3.19 , we can not generate the simplest-threshold function by spectral translation, but minterm-interchange gives a very simple realisation (Figure 3.19.b)

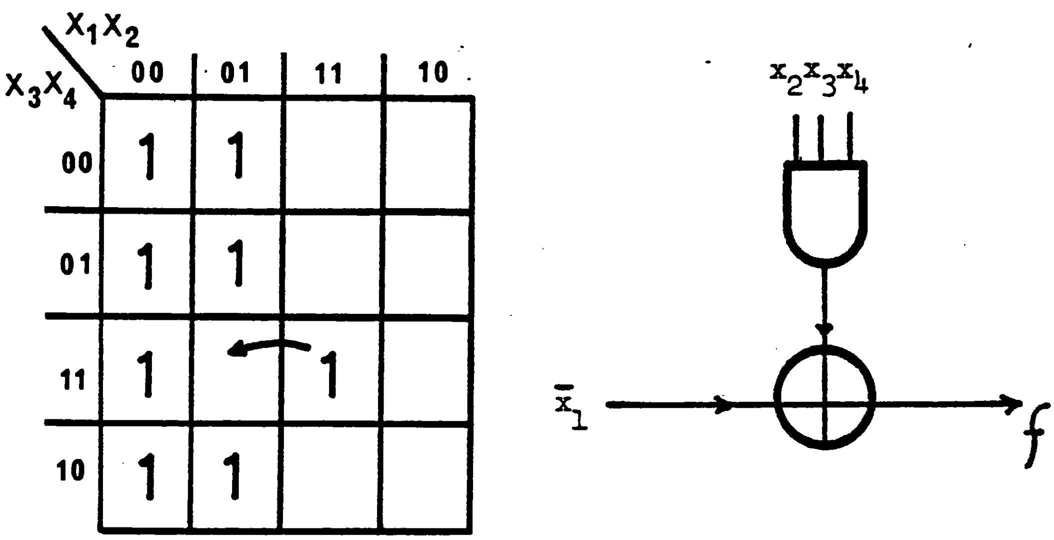


Figure 3.19 An example which shows minterm-interchange adventage

Since one of the methods has advantages over the other according to the given function,by appropriately combining these two methods we may arrive at an optimal realisation.There are two possible ways of combining these approaches,namely:

1) First spectral translation and then minterm-interchange

2) First minterm-interchange and then spectral translation.

Both combinations have the circuit structure shown in Figure 3.20.

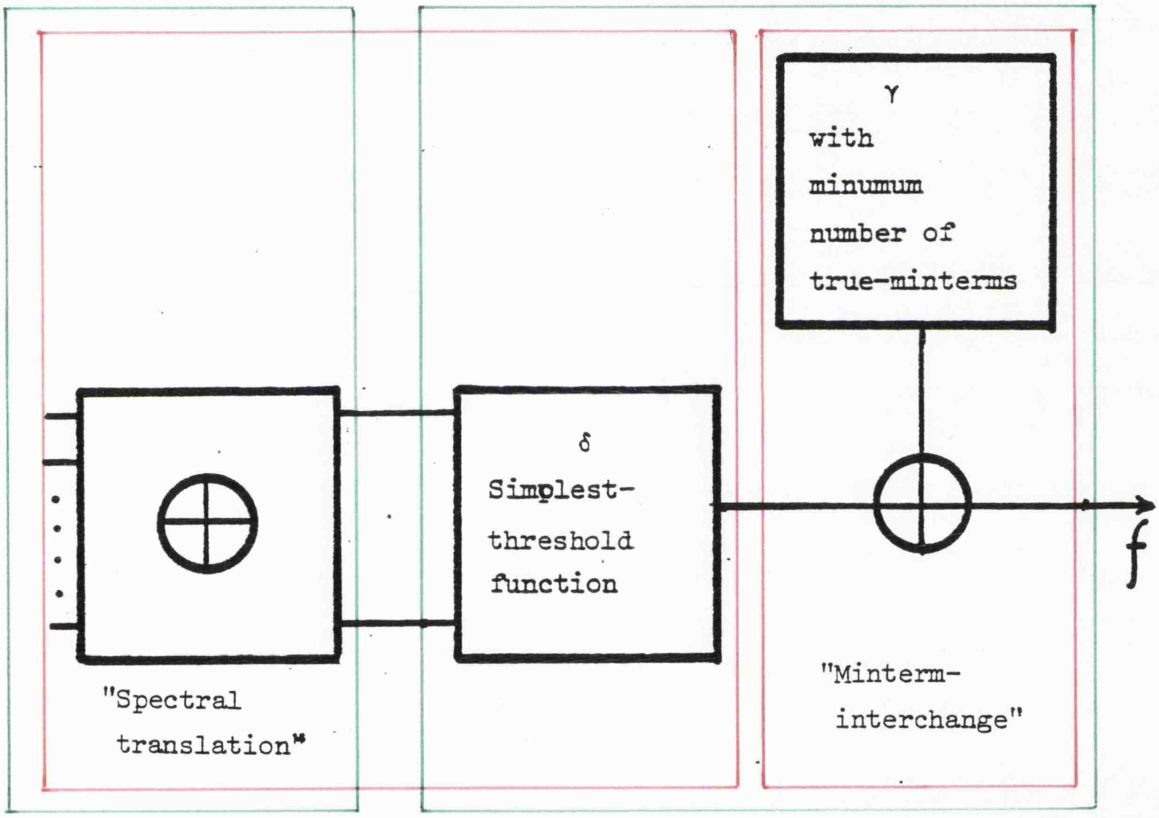


Figure 3.20 Circuit structures of the two combined techniques

If the first-order spectral coeffecients of f are

a) all zero

or b) k number of possible true-minterms in γ is a reasonable high value when compared with u ,

then applying spectral translation first would be useful.This is because by using the spectral translation we can generate high-value-first-order spectral coeffecients which reduces the value of k for our minterm-interchange design purpose.Therefore,the relationship

$$k > \frac{u}{2} \qquad \ldots 3.12$$

has been chosen as a criterion for the sequence of the methods.That is if $k > (u/2)$ we choose spectral translation first (perhaps repeated until $k \leqslant (u/2)$) and then minterm-interchange.The example below illustrates such a case.

Example 3.8 For the function in Figure 3.21 the possible number of true-minterms in Υ is $k = 2.\frac{16-4}{4} = 6$ and satisfies the criterion i.e. $k=6 > \frac{u}{2} = 2$. $R_{24} \longleftrightarrow R_2$ spectral translation gives function f' (Figure 3.21.b).Then by the $m_9 \longleftrightarrow m_{13}$ minterm-interchange we obtain the circuit in figure 3.21.c. The conventional two-level realisation is also shown for comparison in Figure 3.21.d.

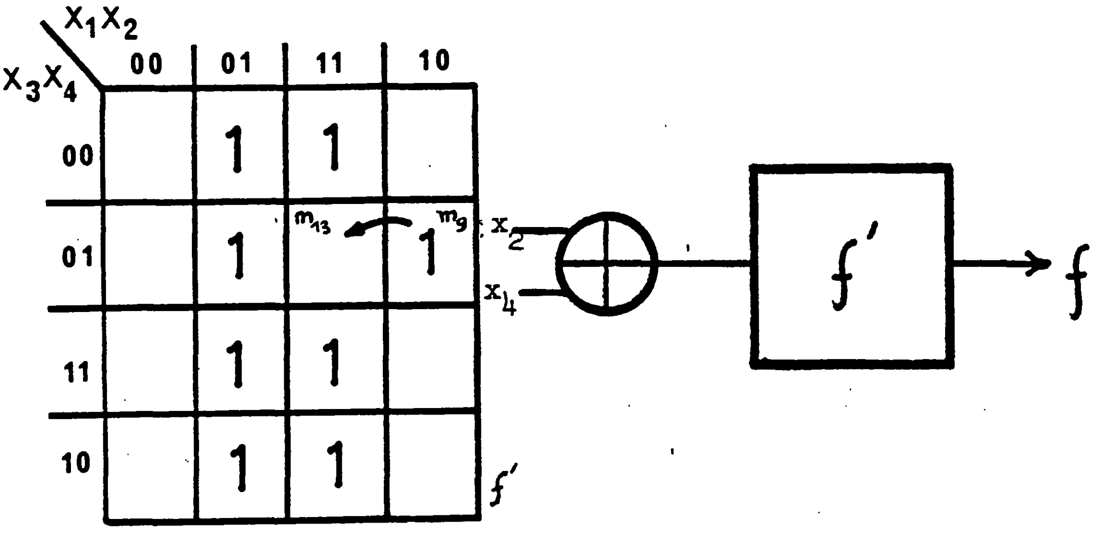
X_3X_4 \ X_1X_2	00	01	11	10
00		1	1	
01	1		1	
11	1			1
10		1	1	

f

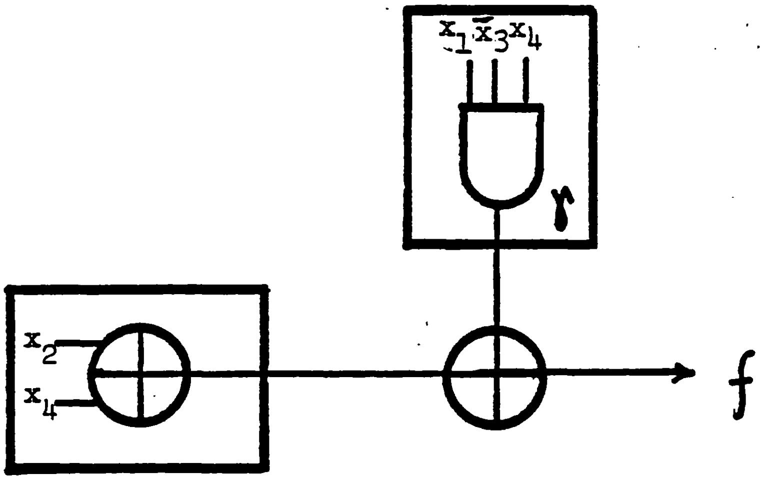
Spectrum of f:

R_0	R_1	R_2	R_3	R_4	R_{12}	R_{13}	R_{14}
0	0	4	0	0	-4	0	0
R_{23}	R_{24}	R_{34}	R_{123}	R_{124}	R_{134}	R_{234}	R_{1234}
4	12	0	-4	4	0	-4	4

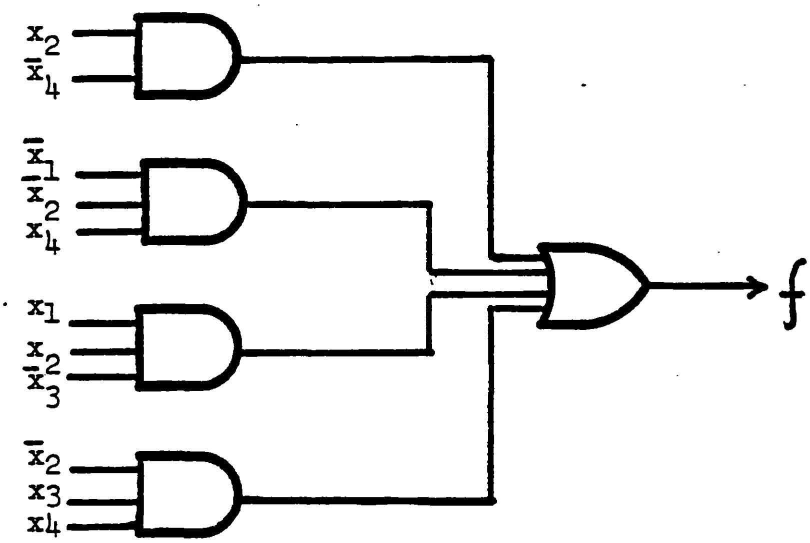
(a) The Boolean function to be designed and its spectrum.



(b) Application of spectral translation to f.



(c) Spectral translation followed by minterm-interchange applied to f'.



(d) Conventional two-level design of the function f.

Figure 3.21 An example which shows the combined method of spectral translation and minterm-interchange

Now we shall consider another combined technique.At the fourth step of the procedure described in 3.2.3 we may not find any $\underset{\sim}{S}_d^{(c)}$ which is equal to $\underset{\sim}{S}_\delta$,after trying $\binom{\ell}{k/2}$ number of possible $\underset{\sim}{C}_d$ combination matrices.In this case k/2 minumum possible-minterm -pair-interchange is not adequate to obtain the simplest-threshold function spectrum $\underset{\sim}{S}_\delta$,but it might be adequate to obtain a simplest-embedded-threshold function to which spectral translation can be applied. A problem arises about which k/2 minterm-pair should be interchanged in order to obtain a simplest-embedded-threshold function. Z_d,which is defined by equation 3.13 ,may be used as a criterion to resolve this problem.

$$Z_d = \underset{\sim}{1}^t \left[\underset{\sim}{S}_\delta^{(c)} - \underset{\sim}{S}_d^{(c)} \right] \qquad \ldots 3.13$$

where

$\underset{\sim}{1} = \begin{bmatrix} 1 & 1 & \ldots 1 \end{bmatrix}^t$

$\underset{\sim}{S}_\delta^{(c)}$ = Canonic spectrum of simplest-threshold function (see Appendix A)

$\underset{\sim}{S}_d^{(c)}$= Canonical spectrum of the minterm-interchanged function corresponding to particular $\underset{\sim}{C}_d$ combinational matrix.

The small value of Z_d means most of the spectral coeffecients of particular $S_d^{(c)}$ and S_δ are equal.Thus we can choose the $\underset{\sim}{C}_d$ or minterm-interchange pairs (i.e. γ) which correspond to minumum Z_d .These particular minterm-interchange pairs may be define a δ function to which we can apply minumum spectral translation operation to obtain the simplest-threshold function δ.The example below illustrates such a case.

Example 3.9. Design the function in Figure 3.22 .This function is not an embedded-threshold function.So we will apply minterm-interchange first.

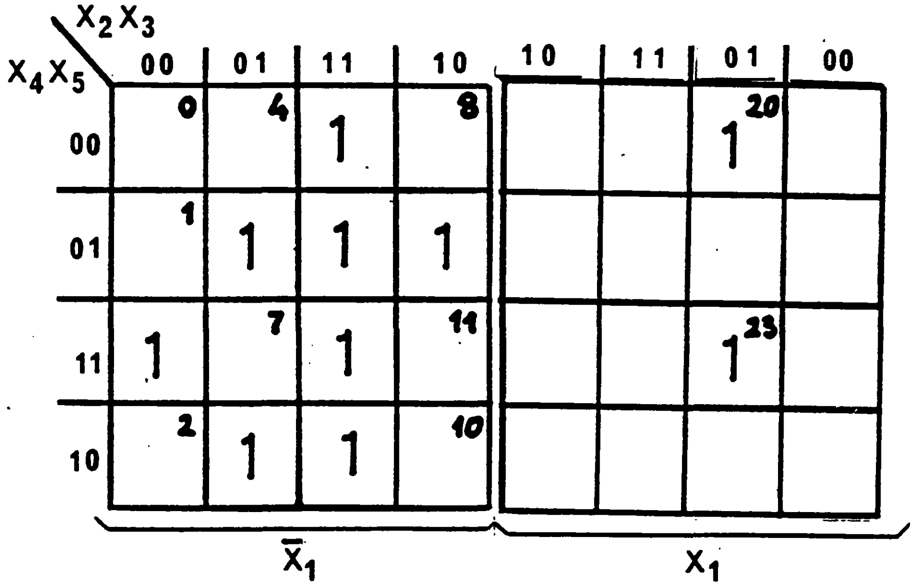


Figure 3.22. A function to which spectral translation cannot usefully be applied first for our decomposition purpose,that is $k = 4 < (u/2) = 5$.

Step 1

$u = 10 \quad , \quad n = 5$

$10 < 2^{n-1} = 2^4 = 16$ we will design f.

Step 2

Spectrum of f $\quad \underset{\sim}{S}_f = \begin{bmatrix} -12 & 0 & 12 & 0 & 4 \\ R_1^f & R_2^f & R_3^f & R_4^f & R_5^f \end{bmatrix}^t$

Canonic form of $\underset{\sim}{S}_f$: $\underset{\sim}{S}_f^{(c)} \begin{bmatrix} 12 & 12 & 4 & 0 & 0 \\ R_1 & R_2 & R_3 & R_4 & R_5 \end{bmatrix}^t$

Spectrum of simplest threshold function δ $\quad \underset{\sim}{S}_\delta = \begin{bmatrix} 20 & 12 & 4 & 4 & 0 \\ R_1' & R_2' & R_3' & R_4' & R_5' \end{bmatrix}$

$$\frac{k}{2} = \frac{R_1' - R_1}{4} = \frac{20 - 12}{4} = 2$$

The corresponding spectral value in S_f is $R_1^f = -12 < 0$. There are $(k/2) = 2$ exact true-minterms in the $x_1 = 1$ space which are:

$m_{20} = (1\ 0\ 1\ 0\ 0)$; $m_{23} = (1\ 0\ 1\ 1\ 1)$.

The true-minterm matrix X :

$$X = \sum_{i=1}^{2} M_i = \begin{array}{c} M_{20} \\ \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \end{array} + \begin{array}{c} M_{23} \\ \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \end{array} = \begin{bmatrix} -2 \\ 2 \\ -2 \\ 0 \\ 0 \end{bmatrix}$$

Step 3

$R_1 = -12 < 0$

$\ell = 2^{n-1} - (n- \frac{k}{2}) = 2^4 - (10 - 2) = 16-8 = 8.$

There are $\ell = 8$ false-minterms in the $x_1=0$ space which are:

$m_0 = (0\ 0\ 0\ 0\ 0)$; $m_1 = (0\ 0\ 0\ 0\ 1)$; $m_2 = (0\ 0\ 0\ 1\ 0)$

$m_4 = (0\ 0\ 1\ 0\ 0)$; $m_7 = (0\ 0\ 1\ 1\ 1)$; $m_8 = (0\ 1\ 0\ 0\ 0)$

$m_{10} = (0\ 1\ 0\ 1\ 0)$; $m_{11} = (0\ 1\ 0\ 1\ 1)$.

The false matrix Q therefore is:

$$Q = \begin{array}{c} \begin{array}{cccccccc} M_0 & M_1 & M_2 & M_4 & M_7 & M_8 & M_{10} & M_{11} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \end{array}$$

Step 4

$$d = \binom{\ell}{k/2} = \binom{8}{2} = \frac{8!}{6!\,2!} = \frac{7.8}{2} = 32$$

$$\underset{\sim}{Y} = \underset{\sim}{S_f} + 2\underset{\sim}{X} = \begin{bmatrix} -12 \\ 0 \\ 12 \\ 0 \\ 4 \end{bmatrix} + 2 \begin{bmatrix} -2 \\ 2 \\ -2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -16 \\ 4 \\ 8 \\ 0 \\ 4 \end{bmatrix}$$

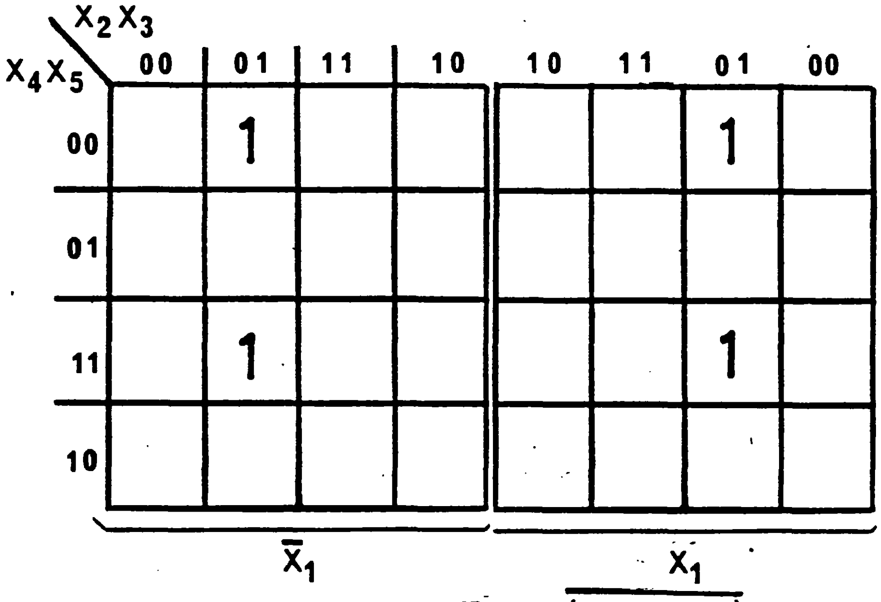
None of 32, C_d combination matrices gives $\underset{\sim}{S_d^{(c)}} = \underset{\sim}{S_\delta}$;therefore we should apply a combined method. One of the $\underset{\sim}{C_d}$ which gives the minumum Z_d is (0 0 0 1 1 0 0 0), and corresponding S_1 is :

$$\underset{\sim}{S_1} = \underset{\sim}{Y} - 2Q\underset{\sim}{C_1} = \begin{bmatrix} -16 \\ 4 \\ 8 \\ 0 \\ 4 \end{bmatrix} - 2 \begin{array}{c} \begin{matrix} M_0 & M_1 & M_2 & M_4 & M_7 & M_8 & M_{10} & M_{11} \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -20 \\ 12 \\ 4 \\ 4 \\ 0 \end{bmatrix}$$

and the other spectra ,and the criterion defined by equation 3.13 are as follows:

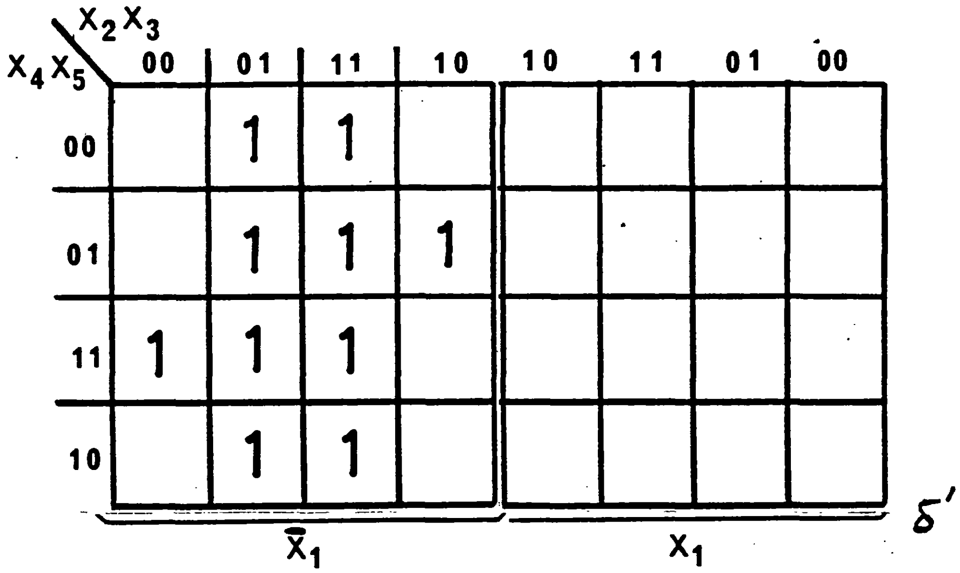
$$\underset{\sim}{S_d^{(c)}} = \begin{bmatrix} 20 \\ 12 \\ 4 \\ 0 \\ 0 \end{bmatrix} \qquad \underset{\sim}{S_\delta} = \begin{bmatrix} 20 \\ 12 \\ 4 \\ 4 \\ 0 \end{bmatrix} \quad Z_d = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 20 \\ 12 \\ 4 \\ 4 \\ 0 \end{bmatrix} - \begin{bmatrix} 20 \\ 12 \\ 4 \\ 0 \\ 0 \end{bmatrix} \right\} = 4 .$$

Now, γ and δ' ,which might be simplest-embedded-threshold,are determined as in Figure 3.23.a and b.



$$Y = \bar{x}_2\, x_3\, \overline{(x_4 \oplus x_5)}$$

(a) Changer function obtained by minterm-interchange.



(b) The function obtained by minterm-interchange.It is not a simplest-thresholdbut by spectral translation it can be made so.

Figure 3.23. Application of minterm-interchange to the function f.

Spectrum of δ' is :

R_0	R_1	R_2	R_3	R_4	R_5	R_{12}	R_{13}	R_{14}	R_{15}	R_{23}	R_{24}	R_{25}	R_{34}
12	-20	0	12	0	4	0	12	0	4	0	4	0	0

R_{35}	R_{45}	R_{123}	R_{124}	R_{125}	R_{134}	R_{135}	R_{145}	R_{234}	R_{235}	R_{245}	R_{345}	R_{1234}
4	0	0	4	0	0	4	0	4	0	-4	0	4

R_{1235}	R_{1245}	R_{1345}	R_{2345}	R_{12345}
0	-4	0	-4	-4

By $R_{24} \longleftrightarrow R_2$ spectral translation on δ' we can obtain the first-order-spectral coeffecient matrix which is equal to $\underset{\sim}{S}_\delta$. That is,there will be a two-input exclusive-OR gate infront of the circuit δ' .

Step 5

The first-order spectrum of δ' after the spectral translation

$R_2 \longleftrightarrow R_{24}$ $S_{(\delta')}^*$ is :

$$\underset{\sim}{S}_{\delta'} = \begin{bmatrix} -20 & 4 & 12 & 0 & 4 \end{bmatrix}^t .$$

The first-order spectrum of the simplest-threshold function and the corresponding δ^* are

$$\underset{\sim}{S}_\delta = \begin{bmatrix} 20 & 12 & 4 & 4 & 0 \end{bmatrix}^t \qquad \delta^* = x_1(x_2 + x_3 x_4).$$

Necessary NP manipulations on δ are:

$$x_1 \longleftrightarrow \bar{x}_1 \quad ; \quad x_2 \longleftrightarrow x_3 \quad ; \quad x_4 \longleftrightarrow x_5$$

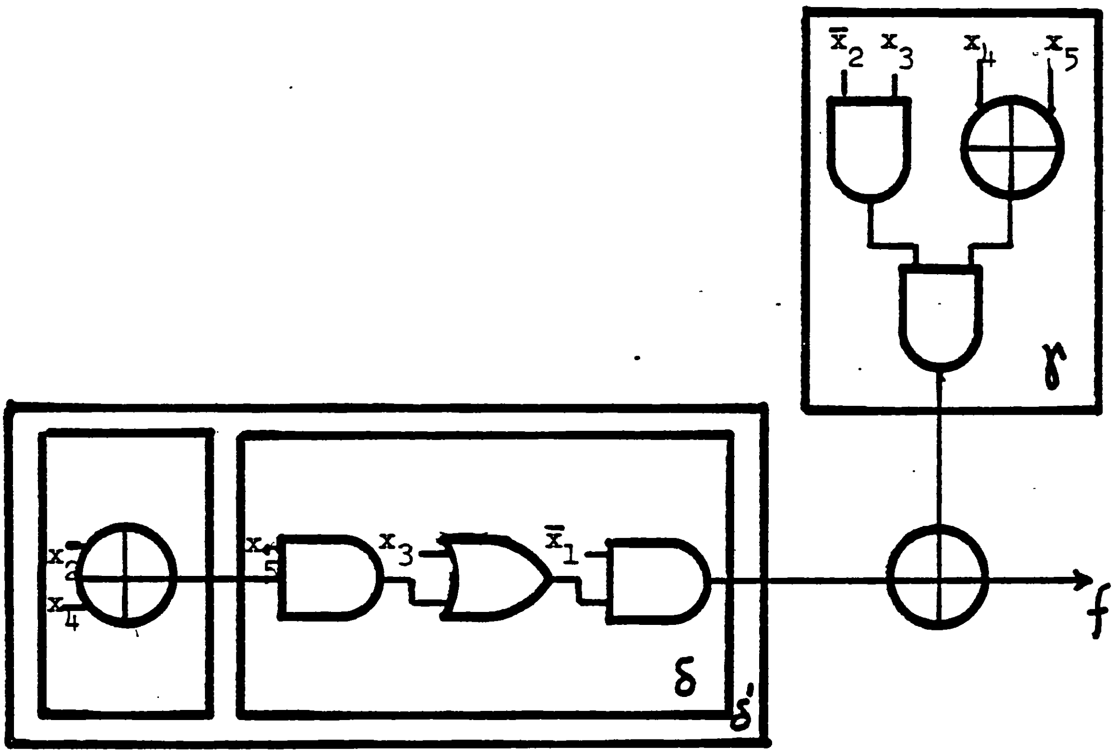


Figure 3.24 The combined realisation:minterm-interchange first and then spectral traslation.

The final simplest-threshold function is ;

$$\delta = \bar{x}_1 (x_3 + x_2 x_5) .$$

The realisations of combined method and conventional method are shown in Figure 3.24 and 3.25 respectively.

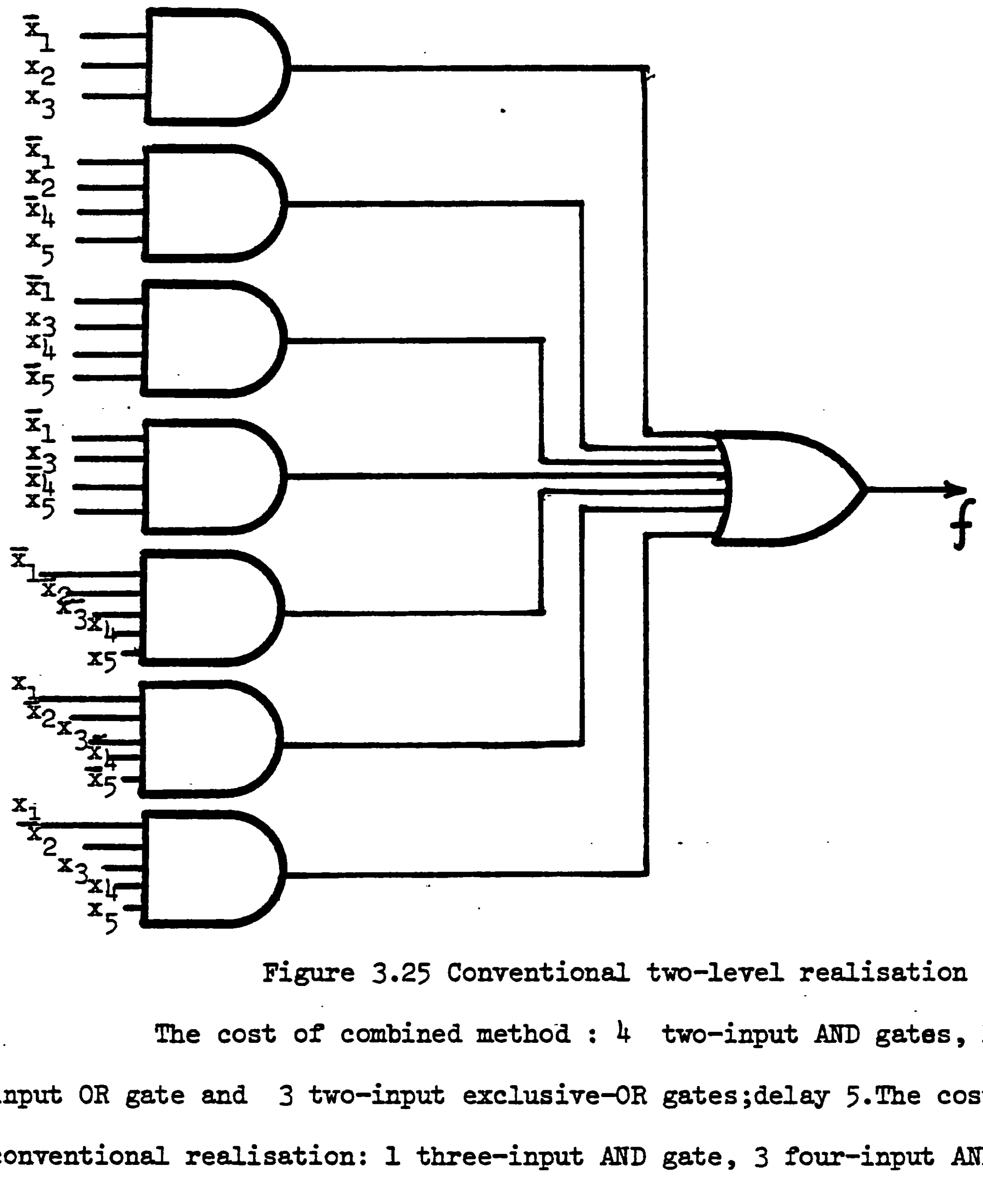
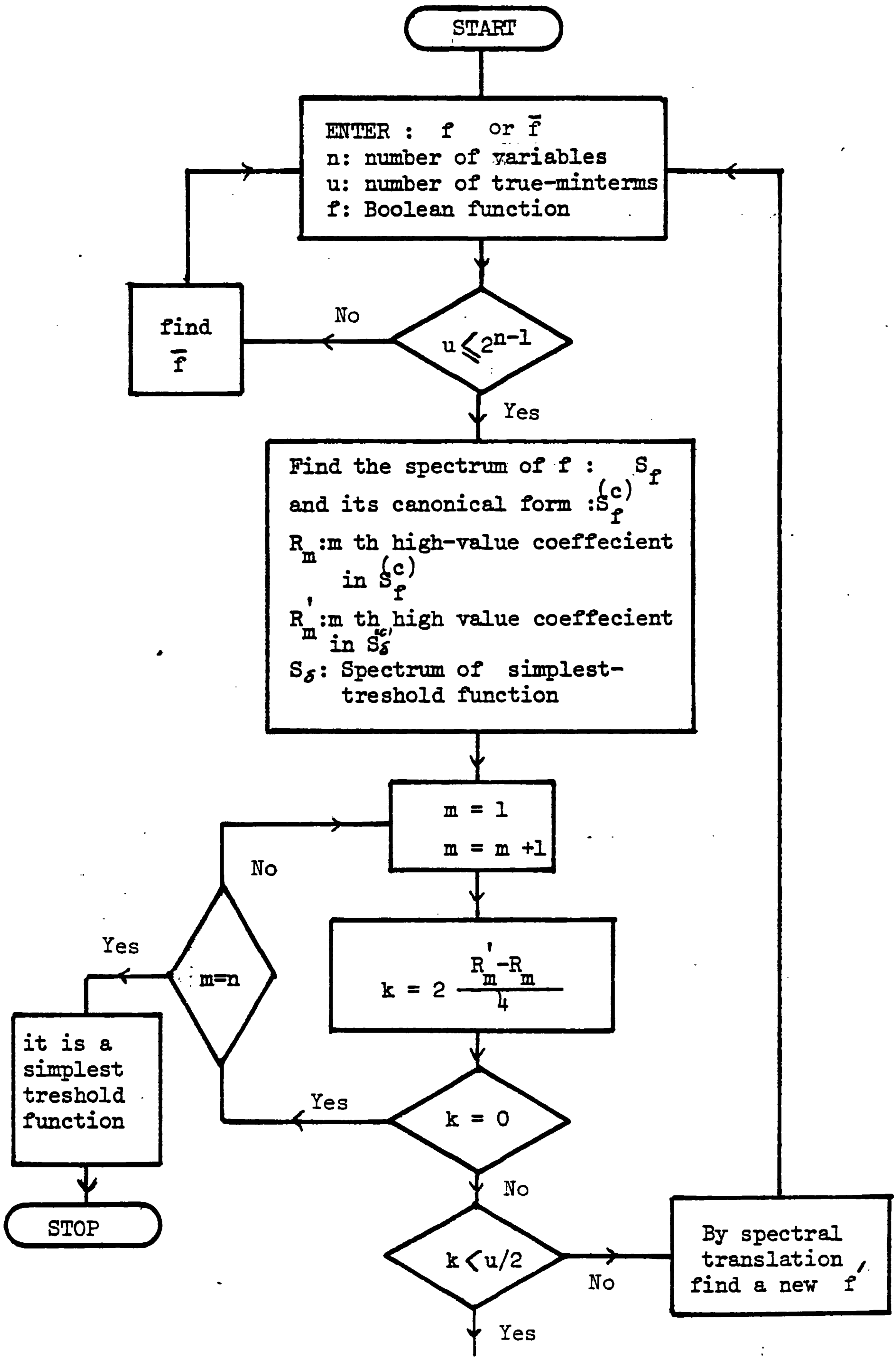
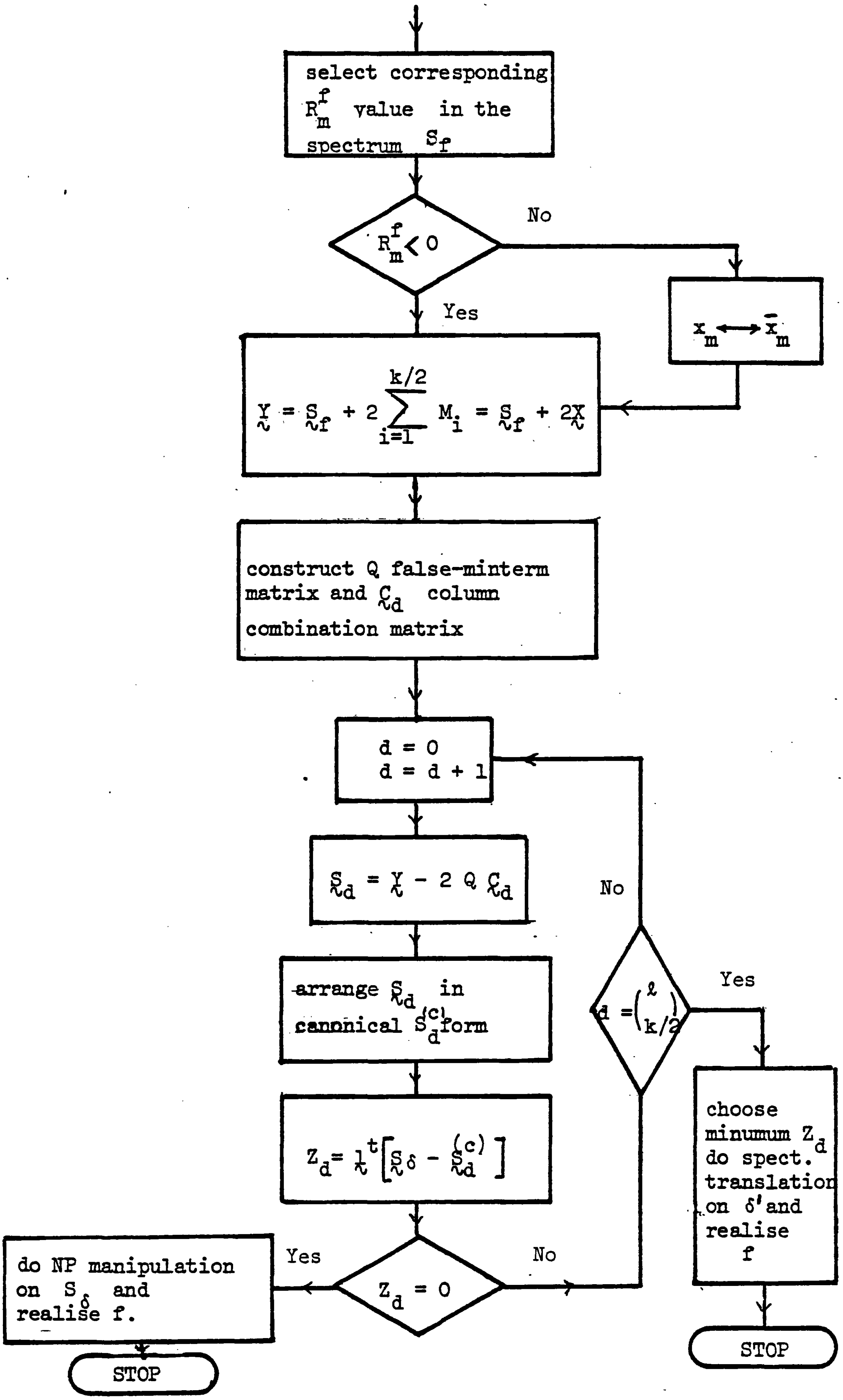


Figure 3.25 Conventional two-level realisation of f.

The cost of combined method : 4 two-input AND gates, 1 two-input OR gate and 3 two-input exclusive-OR gates;delay 5.The cost of conventional realisation: 1 three-input AND gate, 3 four-input AND,and 3 five-input AND gates and 1 seven-input OR gate;delay 2. NOT gates were not taken into account.

A flow chart is given below for minterm-interchange realisation method and combined methods of minterm-interchange and spectral translation.





select corresponding R_m^f value in the spectrum S_f
$R_m^f < 0$
No
Yes
$x_m \longleftrightarrow \bar{x}_m$
$\underset{\sim}{Y} = \underset{\sim}{S}_f + 2 \sum_{i=1}^{k/2} M_i = \underset{\sim}{S}_f + 2\underset{\sim}{X}$
construct Q false-minterm matrix and $\underset{\sim}{C}_d$ column combination matrix
d = 0
d = d + 1
$\underset{\sim}{S}_d = \underset{\sim}{Y} - 2\,Q\,\underset{\sim}{C}_d$
No
arrange $\underset{\sim}{S}_d$ in canonical $S_d^{(c)}$ form
$d = \binom{\ell}{k/2}$
Yes
$Z_d = \underset{\sim}{1}^t \left[\underset{\sim}{S}_\delta - \underset{\sim}{S}_d^{(c)}\right]$
choose minumum Z_d do spect. translation on δ' and realise f
do NP manipulation on S_δ and realise f.
Yes
$Z_d = 0$
No
STOP
STOP

3.4. Conclusion

So far we have applied minterm-interchange operations alone, and also the combination of minterm-interchange and spectral translation operations together, to the problem of combinational circuit design. Basically these are exclusive-OR decompositions. One of the functions of the decomposition has been chosen as a simplest-threshold function with the same number of true (or false) minterms as the original function f. The reasons for doing this are that firstly they are easy to design and secondly they are universal, namely for any given n variables and u true-minterms the circuits which realise these functions are the same excepting NOT gates and input permutations. Because of these properties simplest-threshold functions could be produced as integrated circuits and used as general-purpose gates. Thirdly, simplest-threshold functions are easy to deal with in the spectrum domain. They are uniquely defined by the n+1 spectral coeffecients which correspond to the Chow parameters.

The other function of this decomposition has the minumum possible number of true-minterms, which is statistically a lower cost function than the original function according to the entropy-cost curve (see Figure 3.9).

In general, for the Boolean functions which are close to the central area of the entropy-cost curve, and have an above-average cost, the possibility that this technique may give a better design than the conventional design techniques is high. However if the above-average cost functions have large true/false minterm ratios (they have extreme values of u on the entropy-cost curve) then the possibility of having a better realisation by this technique is very low. This result can clearly be seen from the entropy-cost curve (see Figure 3.9.a,b,c).

Although the minterm-interchange technique sometimes does-not give a better solution ,it is still worth considering this technique as an alternative to the conventional designs,under the above mentioned conditions.

At the end of the minterm-interchange desgn procedure there may be more than one γ functions with the same number of true-minterms in them. In our case we have chosen any one of these possible γ functions functions .More research is needed to choose the best γ function(s).

The idea of exclusive-OR decomposition may also be useful to design multiple-output Boolean functions.Consider two Boolean functions having the same number of true-minterms,then clearly one function can be generated from the other by a simple minterm-interchange operation.

REFERENCES

1. KOHAVI, Z. "Switching and Finite Automata Theory " McGraw-Hill., 1978.

2. GIVONE , D.D. " Introduction to Switching Circuit Theory" McGraw-Hill ,1970.

3. LEWIN,D. " Logical Design of Switching Circuits " Nelson , 1974.

4. MUKHOPADHYAY , A. and SCHMITT,G. "Minimisition of Exclusive-OR and Logical Equivalence Switching Circuits " Trans. IEEE C-19, p.p. 132-140, Feb.1970.

5. REED,I.S. " A Class of Multiple Error Correcting Codes and Decoding Scheme " Trans. IRE, IT 4,p.p.38-49,1954.

6. MULLER,D.E. " Application of Boolean Algebra to Switching Circuit Design and Error Correction " Trans. IRE EC-3,p.p.6-12,Sep.1954.

7 EDWARDS, C.R. " The Application of the Rademacher-Walsh Transform to Boolean Function Classification and Threshold Logic Synthesis" Trans. IEEE , C-24 ,p.p. 48-62, Jan.1975.

8. EDWARDS,C.R. " The Design of Easily Tested Circuits Using Mapping and Spectral Techniques " ; IERE Radio and Electronic Engineer, Vol.47,p.p.321-342,July 1977.

9. KARPOVSKY,M.G. " Finite Orthogonal Series in the Design of Digital Devices " John Wiley. 1976.

10. DAVIO,M. ; DESCHAMPS,J.P. ; THAYSE,A. " Discrete and Switching Functions " McGraw-Hill ,1978.

11. SHANNON,C.E. " The Synthesis of two-terminal Switching Functions" Bell Sys. Tech. Jour. Vol.28 ,p.p. 59-98, Jan. 1949.

12. MULLER,D.E. " Complexity in Electronic Switching Circuits " IRE Trans. Electron. Comput. Vol.EC-5,p.p.15-19 ,March 1956.

13. LUPANOV,D.B. " On a method of Network Synthesis " Izv. Vuzov. Rodiofizika No.1,p.p.43-45,1958.

14. WINOGRAD ,S. " On the Time Required to perform Computer Operations" Ph.D. Thesis,University of New York,March 1967.

15. SPIRA,P. " Computation Times of Arithmetic and Boolean Functions in (d,r) Circuits " Trans. IEEE Comput, C-22 p.p. 552-555,1973.

16. DAVIO,M and QUISQUATER,J.J. " Complexity of Discrete Functions MBLE Internal Report R 292,1975.

17. YABLONSKY,S.W. "On Algorithmic Obstacles to the Synthesis of minimal Contact Networks " Problemy Kibernetiki No.2,p.p.75-121 1959.

18. KELLERMAN,E. " A Formula for Logical Network Cost " Trans.IEEE C-17 No.9,p.p.881-884, Sept. 1968.

19. COOK,R.W. and FLYNN,M.J. " Logical Network Cost and Entropy " Trans. IEEE C-22 No. 4,p.p. 823-826,Sept. 1973.

20. HELLERMAN ,L. " A Measure of Computational Work " Trans.IEEE C-21 No.5,p.p.439-446 , May 1972.

21. MILETO,F. and PUTZOLU, G. " Statistical Complexity of Algorithms for Boolean Function Minimasition " J.ACM Vol.12 No.3 p.p.364-375,July 1965.

22. SHOLOMOV,L.A. " Complexity Criteria for Boolean Functions" Problemy Kibernetiki No.17,1966.

23. HURST,S.L. " The Logical Processing of Digital Signals " Crane Russak,New York 1978.

24. CHOW,C.K. " On the Characterisation of Threshold Functions" Proc. IEEE Symp. on Switching Theory and Logic Design,p.p. 34-38, 1961.

25. WINDER,R.O. " Chow Parameters in Threshold Logic " Journal of the Association for Computing Machinery,Vol.18 No.2, p.p. 265-289, April 1971.

CHAPTER 4. THE APPLICATION OF MINTERM-INTERCHANGE TO THE STATE ASSIGNMENT PROBLEM IN SEQUENTIAL MACHINE DESIGN

CHAPTER 4. THE APPLICATION OF MINTERM-INTERCHANGE TO THE STATE ASSIGNMENT PROBLEM IN SEQUENTIAL MACHINE DESIGN

The notations we will be using in this Chapter are as follows:

$x_1, x_2, \ldots x_\mu$	input variables; μ :number of input variables
$y_1, y_2, \ldots y_\nu$	state variables; ν :number of state variables
$Y_i(x_1 \ldots x_\mu, y_1 \ldots y_\nu)$	next-state functions $1 \leqslant i \leqslant \nu$
A,B,C,...	states (always shown by capital letters)
I,II,...	inputs (always shown by Roman numbers)
2^ν	number of states
2^μ	number of input signals
$n = \mu + \nu$	number of independent variables

4.1. Introduction

One of the main and also most complex problem in designing sequential machines is " state assignment ",namely assigning binary coding to the states,or sometimes to the inputs,of a given sequential machine so that the necessary combinational part of the designed circuit may be minimal.In general,the state-assignment techniques for synchronous and asynchronous sequential machines are different. For example in asynchronous machines problems due to hazards arise.In this thesis we will be dealing with the state assignment of synchronous sequential machines.

Minimisation of the combinational part of a sequential machine depends on the type of the memory element used in the synthesis.If type D (delay) memory elements are used then the minimisation correspond to the simplification of the next-state functions which are the inputs of the delay type of memory elements.If type JK (or SR) memory is used instead of delay elements,then the minimisation corresponds to the simplification of the J,K (or S,R) input functions which can be obtained from the next state functions by the definition-equation of these type of memory elements.

In the light of the above,the state assignment problem can be reduced to the complexity(or cost) concept of Boolean functions under certain conditions which are determined by the design techniques used. This complexity concept depends upon the state assignment technique.

There are two general approaches to the problem of finding economical internal state codes for a sequential machine.Each of these approaches interprets the complexity of a Boolean function differently.Let us examine the development of these two general techniques..

i) the first state assignment approach:

This approach was initiated by Hartmanis and Stearns[4,5] (1961). The fundamental idea in their studies is to find methods for choosing an assignment in which each next-state function depends on as few state variables as possible.Their method relies on the concept that the complexity of a Boolean function with a reduced number of independent variables is better than the one with a higher number of independent variables.The main tools used in their study are the state partition with the substition property,and partition pair algebra on the set of states of a sequential machine.

Curtis [6] and Karp [7] (1962) extended the work of Hartmanis and Stearns,to cover sequential machine state assignment techniques with reduced dependency of state variables.Later Kohavi [8] (1964) presented a method of obtaining,for any given machine M,an equivalent M' which has a partition with the substition property or partition pairs.Finally Weiner and Smith [9] gave an algorithm based upon the theory developed by Hartmanis and Stearns.In all the above mentioned papers only the delay type of memory element was considered.

However,in 1967 Harlow and Coates [10] and in 1969 Curtis[11,12] generalized the known techniques to different type of memory elements such as T " trigger " and JK memory elements.

ii) The second state assignment approach:

So far we have noted the development of state assignment techniques which are concerned with the reduction of the dependency of the state variables.Now we will review the second approach which was initiated by Armstrong [13] in 1961. This method is ,in general, based on the idea of increasing the order of the possible minterm cubes by assigning optimal binary codes to the states. In 1964 Dolotta and Mc Cluskey [14] developed a " scoring " technique to simplfy the next-state functions.Later Torng [15] gave a further technique based on the " cost of an elementary assignment ". Finally,in 1972, Story,Harrison,and Reinhard [16] developed an optimal state assignment technique,defining the " cost " of a Boolean function as the number of AND , OR gates. They also applied this algorithm for sequential machine design with JK memory elements.

In this thesis we will only consider this second state assignment approach.

4.2 Average Complexity (Entropy) of Boolean Functions and State Assignment

As was emphasized in section 4.1,the role of complexity of a Boolean function has a major importance in the state assignment problem of sequential machine design as well as the design of combinational circuits. Recently a state assignment algorithm was given by Lala [17] ,further interpreted by Edwards and Eris [18] using the average entropy cost defined according to the number of true (or false)minterms in a Boolean function,as defined in Chapter 3.1.1.

Although Lala's method[17] can be applied either to complete or incomplete machines,for the sake of simplicity we will state henceforth the basic steps for complete machines only. Other assumptions are :

a)only delay type elements are used,

b) any additional output functions are not considered.

Lala's method :

step 1)

By inspection select $2^{\nu-1}$ states for which the total number of appearances in the state table is minumum.These states, which constitute a " block " are assigned y_1= I and rest of the states are assigned $y_1 = 0$. This will mean that the next-state function Y_1 will have a minumum number of true-minterms (or maximum number of false-minterms).

step 2)

Repeat step 1 for each block of the previous partition. Continue to this operation until a block containing a single state only is found.At this stage all the states are uniquely defined i.e. coding is completed.

This algorithm effectively maximises (or minimises) the true-minterms in every next-state function.This operation,although it is not mentioned in Lala's paper,forces the next-state function towards the u-extremes of the average entropy cost curve [18], shown shaded in Figure 4.1.

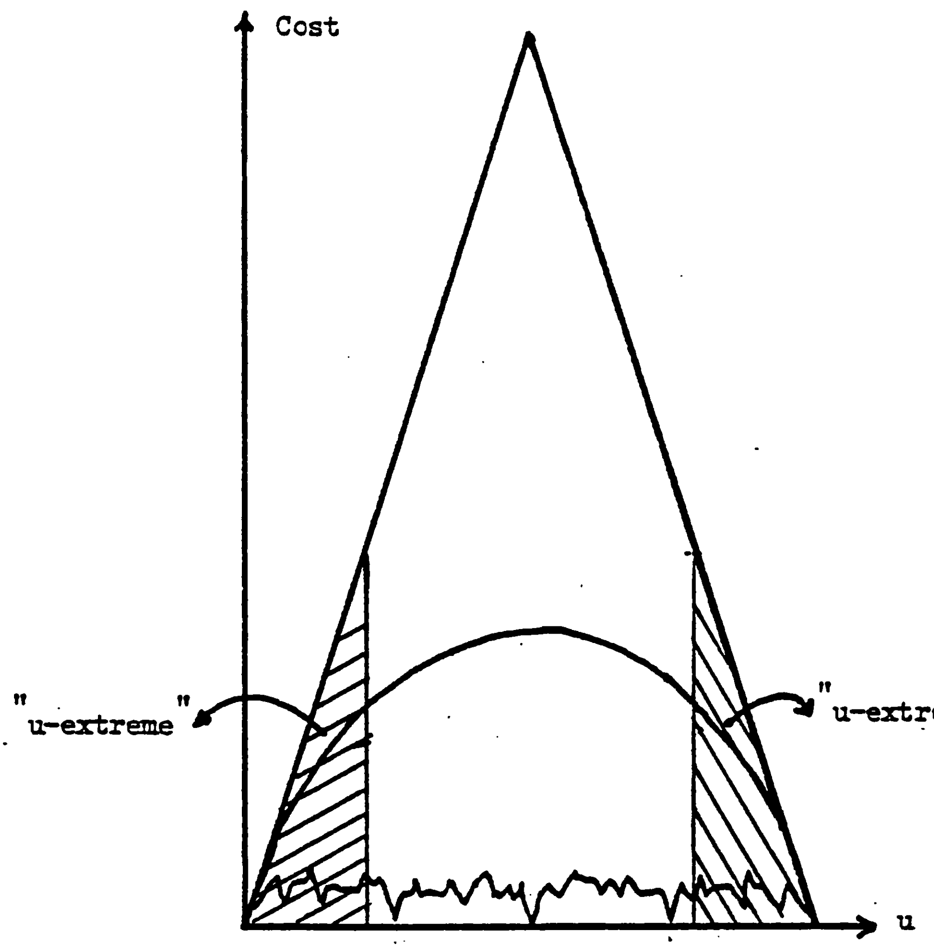


Figure 4.1 Interpretation of Lala's algorithm on the cost curve

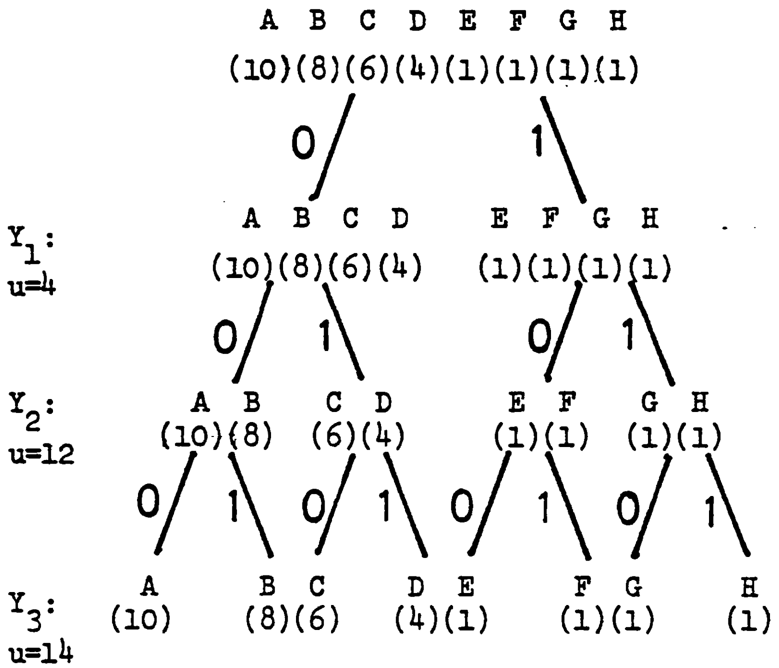
The result of such a distribution is that next-state functions will statistically be simpler than those which are determined by randomly chosen assignments. This method can also be graphically illustrated by a binary tree.

Example 4.1 Apply Lala's algorithm to the machine in Figure 4.2.a. A binary tree is established as in Figure 4.2.b. The numbers underneath the each state show the number of times of appearance of the respective states in the state table. The construction of the binary tree for this example is as folows :

step 1) Let us choose $2^{\nu-1} = 2^{3-1} = 4$ number of states out of $2^{\nu} = 2^3 = 8$ total number of states as (E F G H). The total number of appearances of these states in the state table is 4, which is a

Total number of times of appearances of the states:	$y_1y_2y_3$ \ x_1x_2	I	II	III	IV
10	A	D	D	C	C
1	E	C	C	D	D
1	G	A	A	A	A
4	D	A	B	B	A
6	C	A	A	C	C
1	F	G	B	B	H
1	H	A	B	B	A
8	B	E	B	B	F

Figure 4.2.a Example sequential machine for the application of Lala's state assignment



Resultant State code:

	y_1	y_2	y_3
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

Figure 4.2.b. Binary tree determined by Lala's algorithm for the sequential machine given in Figure 4.2.a.

unique minumum value.We shall assign these states the value of the first state variable $y_1 = 1$,and the rest of the states (A B C D)the value of the first state variable $y_1 = 0$ (see binary tree in Figure 4.2.b). The resultant next-state function Y_1 will have 4 true-minterms $u = 4$ (see figure 4.2. b,c).

step 2) Repeating the previous step to each partition block obtained above,we must choose (C D) states,with the minimal total appearance (6 + 4 = 10),out of the set (A B C D). Similarly the (G H) states ,with the total appearance (1 + 1 = 2) out of the set (E F G H),are chosen.For the latter partition block we could choose any two states,because they always give the same total appearance number. Thus (G H) is chosen arbitrarily. We shall assign (C D G H) states the value of second state variable $y_2 = 1$, and (A B E F) states the value of second state variable $y_2 = 0$.(see binary tree in Figure 4.2.b).The resultant next-state function Y_2 will have 12 true-minterms, $u = 12$,as in Figure 4.2.d. Finally, applying the same principle above we will choose (B D F H) states with the minumum total appearance 14,and give these states the value of the third state variable $y_3 = 1$ (see the binary tree in Figure 4.2.b). For the state codes determined by the binary tree,the resultant next-state function Y_1 with four true-minterms, Y_2 with twelve true-minterms , and Y_3 with fourteen true-minterms are shown in Figure 4.2.c,d,e below.

This algorithm of Lala statistically guarantees the simlicity of next-state functions. However there is no absolute guarantee it will give a good result.

$y_1y_2y_3 \backslash x_1x_2$	00	01	11	10
000				
001	1			1
011				
010				
110				
111				
101	1			1
100				

Figure 4.2.c Next-state function : $Y_1 = \bar{x}_2\bar{y}_2y_3$

$y_1y_2y_3 \backslash x_1x_2$	00	01	11	10
000	1	1	1	1
001				
011				
010				
110				
111				
101	1			1
100	1	1	1	1

fig.4.2.d) $Y_2 = \bar{y}_2\bar{y}_3 + \bar{x}_2y_1y_2$

$y_1y_2y_3 \backslash x_1x_2$	00	01	11	10
000	1	1		
001		1	1	1
011		1	1	
010				
110				
111		1	1	
101		1	1	1
100			1	1

Fig.4.2.e) $Y_3 = x_2y_3 + x_1\bar{y}_2y_3 + x_1y_1\bar{y}_2 + \bar{x}_1\bar{y}_1\bar{y}_2\bar{y}_3$

Figure 4.2 Application of Lala's state assignment method to the given sequential machine

4.3 Minterm-interchange in State Assignment

In this section, in order to obtain an optimal state assignment, we shall consider minterm-interchange operations in basic state functions rather than in next-state functions.

4.3.1 Basic state functions

Definition 4.1 Basic state function :. When a binary code is given to the states of a sequential machine for every individual state, a function can be determined from the state table representing the sequential machine by substituting " 1 " for all the minterm positions in which this individual state appears, and " 0 " for all the minterm positions in which this state does not appear. Such a function called a " basic state function ".

Henceforth we denote the basic state functions as $f_A, f_B, f_C, \ldots$.

Example 4.2 Let us find the basic state functions for the sequential machine given in Figure 4.3.a. The binary codes chosen for the states are A(0 0) ; B (0 1) ; C (1 1) ; D (1 0). The basic state functions related to this sequential machine, with the above assignment, are shown in Figure 4.3.b,c,d.

	$y_1y_2 \backslash x_1x_2$	00	01	11	10
A	00	A	A	B	C
B	01	B	C	A	B
C	11	A	A	D	B
D	10	C	C	A	D

	y_1	y_2
A	0	0
B	0	1
C	1	1
D	1	0

Figure 4.3.a. Example sequential machine and its assignment

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00	1	1		
01			1	
11	1	1		
10			1	

b) Basic state function f_A:

$f_A = \bar{x}_1\bar{y}_1\bar{y}_2 + \bar{x}_1 y_1 y_2 + x_1 x_2 (y_1 \oplus y_2)$

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00			1	
01	1			1
11				1
10				

c) Basic state function f_B

$f_B = x_1\bar{x}_2 y_2 + x_1 x_2 \bar{y}_1\bar{y}_2 + \bar{x}_1\bar{x}_2\bar{y}_1 y_2$

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00				1
01		1		
11				
10	1	1		

d) Basic state function f_C

$f_C = \bar{x}_1 y_1 \bar{y}_2 + \bar{x}_1 x_2 \bar{y}_1 y_2 + x_1 \bar{x}_2 \bar{y}_1 \bar{y}_2$

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00				
01				
11			1	
10				1

e) Basic state function f_D

$f_D = x_1 x_2 y_1 y_2 + x_1 \bar{x}_2 y_1 \bar{y}_2$

Figure 4.3. Example for basic state functions for the given state code

Corollary 4.1. The number of basic state functions for a given sequential machine with a known assignment is equal to 2^{ν} .

Corollary 4.2. Basic state functions are disjoint between themselves.

It is because there cannot be two different states in the same minterm position.

Corollary 4.3. The next-state functions of a sequential machine for a given state assignment, can be expressed as exclusive-OR or AND combination of the basic state functions.

For the sequential machine and the given state assignment in Figure 4.3, the next-state functions Y_1 and Y_2 and their complements $\overline{Y}_1$ and $\overline{Y}_2$ are shown as the combinations of the basic state functions in Figure 4.4.a,b below.

y_1y_2 \ x_1x_2	00	01	11	10
00				1
01		1		
11			1	
10	1	1		1

$Y_1 = f_C + f_D = f_C \oplus f_D$

y_1y_2 \ x_1x_2	00	01	11	10
00	1	1	1	
01	1		1	1
11	1	1		1
10			1	

$\overline{Y}_1 = f_A + f_B = f_A \oplus f_B$

a) Next-state function Y_1 and its complement $\overline{Y}_1$

y_1y_2 \ x_1x_2	00	01	11	10
00			1	1
01	1	1		1
11				1
10	1	1		

$Y_2 = f_B + f_C = f_B \oplus f_C$

y_1y_2 \ x_1x_2	00	01	11	10
00	1	1		
01			1	
11	1	1	1	
10			1	1

$\overline{Y}_2 = f_A + f_D = f_A \oplus f_D$

b) next-state function Y_2 and its complement $\overline{Y}_2$

Figure 4.4 Next-state functions as combinations of basic state functions.

4.3.2 The General Principle for a State Assignment by Minterm-interchange

Let us consider the effect on the basic state functions,and also the next-state fuctions ,of changing the assignment of a given sequential machine.For this purpose we shall follow the example sequential machine given in Figure 4.5.a,with the state assignment A (0 0) ; B (0 1) ; C (1 1) ; D (1 0).Basic state and next-state functions are shown in Figure 4.5.b,c,d,e,f,g.

Example 4.3 Find the next-state functions and their expressions as the combination of the basic state functions for the given state assignment.

	y_1y_2 \ x_1x_2	00	01	11	10
A	00	A	A	B	B
B	01	B	D	B	A
C	11	C	C	D	A
D	10	C	B	A	A

	y_1	y_2
A	0	0
B	0	1
C	1	1
D	1	0

a) Example sequential machine and the given assinment

y_1y_2 \ x_1x_2	00	01	11	10
00	1	1		
01				1
11				1
10			1	1

b) $f_A=\bar{x}_1\bar{y}_1\bar{y}_2+x_1\bar{x}_2y_1+x_1y_1\bar{y}_2$

y_1y_2 \ x_1x_2	00	01	11	10
00			1	1
01	1		1	
11				
10		1		

c) $f_B=x_1\bar{y}_1\bar{y}_2+x_1x_2\bar{y}_1+\bar{x}_1\bar{x}_2\bar{y}_1y_2+\bar{x}_1x_2y_1\bar{y}_2$

y_1y_2 \ x_1x_2	00	01	11	10
00				
01				
11	1	1		
10	1			

d) $f_C = \bar{x}_1\bar{x}_2y_1 + \bar{x}_1y_1y_2$

y_1y_2 \ x_1x_2	00	01	11	10
00				
01		1		
11			1	
10				

e) $f_D = \bar{x}_1x_2\bar{y}_1y_2 + x_1x_2y_1y_2$

y_1y_2 \ x_1x_2	00	01	11	10
00				
01		1		
11	1	1	1	
10	1			

f) $Y_1 = f_C + f_D$

y_1y_2 \ x_1x_2	00	01	11	10
00			1	1
01	1		1	
11	1	1		
10	1	1		

g) $Y_2 = f_B + f_C$

Figure 4.5 The effect of a given assignment to the basic state and next-state functions

Suppose we now interchange the binary codes given to the states B and C, we obtain the new basic-state and next-state functions which are shown in Figure 4.6. These new functions are denoted by adding primes on the original notation.

Let us compare the basic state functions determined by the previous and present state assignments.We can observe that B and C rows have been interchanged in the Karnaugh map of the previous basic state function.That is,corresponding minterms in these rows are interchanged.It is also obvious that number of true-minterms in these

	y_1y_2 \ x_1x_2	00	01	11	10
A	00	A	A	B	B
C	01	C	C	D	A
B	11	B	D	B	A
D	10	C	B	A	A

	y_1	y_2
A	0	0
C	0	1
B	1	1
D	1	0

a) sequential machine for the interchanged assignments of the states B and C

y_1y_2 \ x_1x_2	00	01	11	10
00	1	1		
01				1
11				1
10			1	1

b) $f'_A = f_A = \bar{x}_1\bar{y}_1\bar{y}_2 + x_1\bar{x}_2y_1 + x_1y_1\bar{y}_2$

y_1y_2 \ x_1x_2	00	01	11	10
00			1	1
01				
11	1		1	
10		1		

c) $f'_B = x_1\bar{y}_1\bar{y}_2 + \bar{x}_1\bar{x}_2y_1y_2 + x_1x_2y_1y_2 + \bar{x}_1x_2y_1\bar{y}_2$

y_1y_2 \ x_1x_2	00	01	11	10
00				
01	1	1		
11				
10	1			

d) $f'_C = \bar{x}_1\bar{y}_1y_2 + \bar{x}_1\bar{x}_2y_1\bar{y}_2$

y_1y_2 \ x_1x_2	00	01	11	10
00				
01			1	
11		1		
10				

e) $f'_D = x_1x_2\bar{y}_1y_2 + \bar{x}_1x_2y_1y_2$

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00			1	1
01			1	
11	1	1	1	
10		1		

f) $Y_1^{'} = f_B^{'} + f_D^{'}$

$y_1y_2 \backslash x_1x_2$	00	01	11	10
00				
01	1	1	1	
11		1		
10	1			

g) $Y_2^{'} = f_C^{'} + f_D^{'}$

Figure 4.6 New basic state and next-state functions after interchanging the binary codes of the states B and C.

functions remains invariant when the state assignment changed.

However considering the next-state functions, clearly Y_1 (Y_2) in Figure 4.5 cannot be mapped to $Y_1^{'}$ ($Y_2^{'}$) in Figure 4.6 by row interchange (which correspond to minterm-interchange).This observation is generally true.The effect of changing Y,as above,is twofold:Firstly the basic state function combination for Y and $Y^{'}$ will differ.Secondly,and independently,all basic state functions will change.

From the above example we can conclude that controlling the next-state functions by changing the given assignment is difficult.This is because of the two simultaneous changes in these functions.However the effect of an assignment change on the basic state functions is simply " row interchange"in the Karnaugh map of these functions.Thus as far as basic state functions are concerned,in order to obtain a different state assignment from a known one it is sufficient to interchange the rows of the Karnaugh map by which the basic state functions are defined.

We can now state the principle for a new state assignment technique. Since basic state functions have the above explained advantage over next state functions,our target,if it is possible, is to find basic state functions as simplest-threshold functions *, by changing the binary coding of the states of a given sequential machine.The reasons are as follows:

i) It is essential from the minterm-interchange point of view that we employ basic state functions as opposed to next-state functions,

ii) Since the next-state functions are certain combinations of basic state functions according to the chosen assignment,then the basic state functions can be chosen as simplest-threshold functions (see Appendix A). This means that the next-state functions will , in general, be simple because there is a direct relationship between all minterm cubes in basic state functions and the minterm cubes in the next-state functions.

iii)Each basic state function appears in all next-state functions(to within complementation).This realationship gives us the advantage of using simplest-threshold basic-state functions as common functions for the realisation of next-state functions.

4.3.3 Procedure

The following assumptions are made:

i) start with a reduced state table

ii) the minumum number of memory elements will be used

iii) delay type memory elements will be used

* see Chapter 3.1.2 which details the practical significance of simplest-threshold functions.

iv) sequential machines are complete

v) input coding is already fixed

vi) output functions are not considered,this assumption being made for the sake of simplicity and does not restrict the method.

The procedure will be mainly executed in the $\langle 0\ 1 \rangle$ Boolean domain as opposed to the spectrum domain.Therefore we shall consider the simplest-threshold functions in the Boolean domain.As is explained in Chapter 1.3 by equation 1.24 ,there is a direct relation between first-order spectral coeffecients and the distributions of true minterms,see Appendix A. In order to distribute the true minterms of the basic state functions in such a way that each of these functions becomes a simplest-threshold (if possible),suitable binary codes must be allocated to the states.

Let us combine the two properties of the basic state functions in the state assignment procedure for the method stated in section 4.3.2.They are as follows:

i) When the coding of the states of a sequential machine is changed,the rows of the Karnaugh map of the basic state functions remain invariant but the positions of the rows change.

ii) By changing the state assignment,which corresponds to interchanging the rows in the Karnaugh map of the basic state functions, we may transform the basic state functions to the simplest-threshold functions.

Thus all that is necessary is to change the state assignment in order to obtain each basic state function the same as the related simplest threshold function*,that is both of these functions would have the same

---

* These functions have the same n and u values with the basic state functions .They can be found in Appendix A.

true-minterm distribution in the 0,1 spaces of the independent variables. The procedure will describe the basic state functions by the number of true-minterms in each of the rows of the Karnaugh map of these functions. This is illustrated in the example 4.4. below.

Example 4.4 For the sequential machine in Figure 4.7, the basic state functions are as follows:

	I	II	III	IV
A	A	A	B	B
B	C	C	D	A
C	B	D	B	A
D	C	B	A	A

Figure 4.7 An example of describing basic state functions as rows

Basic state function f_A;

A(2) ; number of A states (true minterms) in the row A.

B(1) : number of A states (true minterms) in the row B.

C(1) : number of A states (true minterms) in the row C.

D(2) : number of A states (true minterms) in the row D.

Basic state function f_B:

A(2) : number of B states (true minterms) in the row A.

B(0) : number of B states (true minterms) in the row B.

C(2) : number of B states (true minterms) in the row C.

D(1) : number of B states (true minterms) in the row D.

Basic state function f_C:

A(0) : number of C states (true minterms) in the row A.

B(2) : number of C states (true minterms) in the row B.

C(0) : number of C states (true minterms) in the row C.

D(1) : number of C states (true minterms) in the row D.

Basic state function f_D:

A(0) : number of D states (true minterms) in the row A.

B(1) : number of D states (true minterms) in the row B.

C(1) : number of D states (true minterms) in the row C.

D(0) : number of D states (true minterms) in the row D.

Since there are 2^{ν} number of state functions,we have to make a decision about which one should be treated first.We will start with a basic state function which has not extreme u values(u being equal to the number of times of the appearance of the related state in the state-diagram).According to the entropy cost curve(see Figure 4.1) these functions are more critical,complexity-wise, than those which have a small number of true-minterms.When the entropy cost curve examined, see Figure 4.1,it can be seen that the difference between the average cost and the maximum cost for the functions which are u-extreme is less than for the functions which are not u-extreme.So the functions which are to be treated first are close to the central area of the entropy cost curve.

Now we can explain the procedure step by step:

step 1)

Find a basic state function with the highest number o f true-minterms.If there is more than one basic state functions with the same number of true-minterms, choose arbitrarily.

step 2)

Give a random state assignment to the states,and find the first-order spectral coeffecients of the chosen basic state function. Since the input assignment is already fixed(assumption v),it follows that the first-order spectral coeffecients of the chosen basic state function,corresponding to the input variables, do not change when the state assignment is changed.The first-order spectral coeffecients corresponding to the input variables must be among the first-order spectral coeffecients of the related simplest-threshold function.If this were not so,the chosen basic state function could not be formed as a simplest-threshold function by simply changing the assignment, that is by interchanging the rows of the Karnaugh map of the basic state function . Thence we continue with another basic state function with the second highest number of true-minterms.

step 3)

In order to transform the chosen basic state function to a simplest threshold function by " row interchange ",the true-minterms of the basic state functions will be distributed in the state variable spaces of the Karnaugh map of the basic state function in such a way that a simplest-threshold function is generated.This is accomplished by means of a binary tree similar to that of section 4.2.Here basic state functions are used instead of next state functions.In this new binary tree the numbers under the states show the number of true-minterms in the corresponding rows of the Karnaugh map of the basic state function.The first partition of the binary tree determines the true-minterm distribution in $y_1 = 0$,1 spaces of the Karnaugh map of the basic state function . Similarly,the second partitioning in the binary tree determines the true-minterm distribution for y_2,etc.

Since our target functions (related to the basic state functions) are simplest-threshold functions, the required true-minterm distribution in $y_1, y_2, .. y_\nu$ spaces of the Karnaugh map of the basic state functions are known from Appendix A. Thus we can arrange the partitions in the binary tree in order to confirm our target functions.

step 4)

Repeat step 3 using the previous binary tree for the possible basic state functions (i.e. those having the second highest number of true-minterms) and continue with the other basic state functions until all the state codings are determined.

Example 4.5 Let us find the state assignment for the sequential machine given in Figure 4.8.

	$y_1y_2y_3$ \ x_1x_2	00	01	11	10
A	000	G	G	G	G
B	001	A	D	D	D
C	011	G	G	C	C
D	010	B	B	E	E
E	110	F	F	C	C
F	111	H	D	D	D
G	101	H	H	B	B
H	100	A	A	E	E

Figure 4.8 Example sequential machine for the new state assignment technique

step 1)

Basic state function f_G has 6 true-minterms.(n=5,u=6)

Basic state function f_D has 6 true-minterms.(n=5,u=6)

Basic state function f_B has 4 true-minterms.(n=5,u=4)

Basic state function f_C has 4 true-minterms.(n=5,u=4)

Basic state function f_E has 4 true-minterms.(n=5,u=4)

Basic state function f_A has 3 true-minterms.(n=5,u=3)

Basic state function f_H has 3 true-minterms.(n=5,u=3)

Basic state function f_F has 2 true-minterms.(n=5,u=2)

This ordering of the basic state functions is also the order in which the basic state functions are to be taken.Since the basic state function f_G has the highest number of true-minterms we start with this function.

step 2)

The first-order spectral coeffecients and also the true-minterm distribution for the related simplest-threshold function are as follows(see Appendix A for n=5 , u=6):

	R_1	R_2	R_3	R_4	R_5
first-order spectral coeffecients:	12	12	4	4	0
true-minterm distribution :	6	6	4	4	3

For a given random state assignment, the absolute values of the first-order spectral coeffecients of the basic state function f_G, which correspond to its input variables, are (4 0).Since these (4 0) coefficients are among the first-order spectral coeffecients of the related simplest-threshold function,then the basic state function f_G is a candidate for a simplest-threshold function.

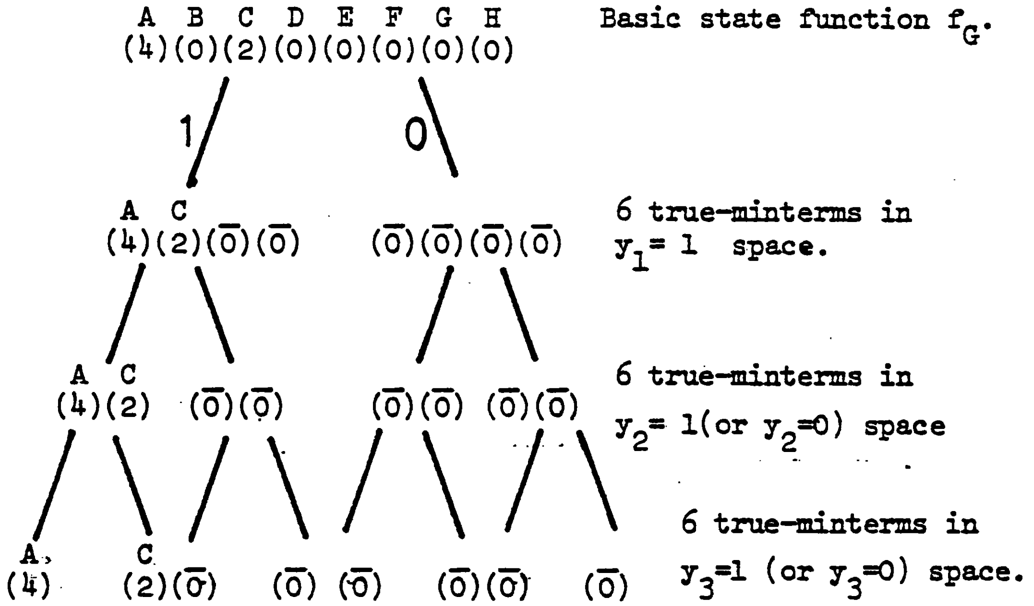
step 3)

The true-minterm distribution in $y_i = 1$ (or $y_i = 0$)space for the related simplest-threshold function is (6 4 4).Using the binary tree we shall try to obtain the same distributions for the basic state function f_G .The binary tree f_G is given in Figure 4.9.a. The first line shows the new description of the basic state function, as illustrated in example 4.4 .The numbers under the each state show the number of true-minterms in the corresponding rows of the Karnaugh map of the basic-state function f_G .As illustrated in Figure 4.9.a ,in order to obtain the same true-minterm distribution for y_1 , y_2 , y_3 independent state variables as the related simplest-threshold function, state A and state C must have the same binary code (" 0 " or " 1 ") for y_1, and similarly for y_2.However a different binary code is required for y_3 .

step 4)

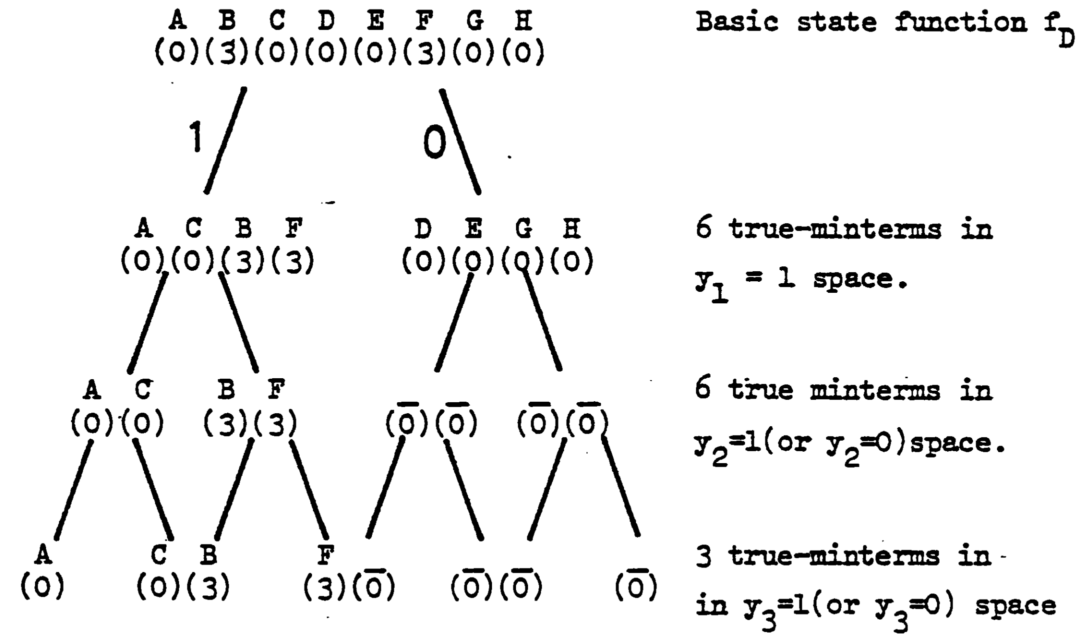
Since the state codes are not determined exactly by the previous incomplete binary tree,we are able to form another basic state function as a simplest-threshold function.Since the basic state function f_D has as many true-minterms as the basic state function f_G, it follows that the related simplest-threshold function to the corresponding state functions f_G and f_D are the same. Although the first-order spectral coeffecients of the basic state function f_D(4 4),which correspond to input variables,are different from those of f_G,they are among the first-order spectral coeffecients of the related δ simplest-threshold function.Therefore we can try to form f_D with the true-minterm distribution (6 6 3),see Figure 4.9.b. B and F states should be given the same code for y_1 and y_2,but a different code for y_3.

The next basic state function we will treat is f_B with four true-minterms.The first-order spectral coeffecients of the related simplest-threshold function are(see Appendix A for n=5 u=4):



first-order spectral coeffecients of
the related simplest-threshold δ : 12 12 4 4 0
and true-minterm distribution : (6) (6) (4)(4)(3)
✓ ✓

Figure 4.9.a Binary tree for the basic state function f_G.



first-order spectral coeffecients of
the related simplest-threshold δ : 12 12 4 4 0
and true-minterm distribution : (6) (6) (4) (4) (3)
✓ ✓

Figure 4.9.b Binary tree covers the basic state functions f_G and f_D.

NOTE: For all figures in 4.9; "✓" shows the minterm distributions which correspond to input variables for a randomly given state assignment.

(8 8 8 0 0)

and the corresponding minterm-distribution:

(4 4 4 2 2).

The first-order spectral coeffecients (0 0) of the basic state function f_B for the input variables x_1, x_2 are among the first-order spectral coeffecients of the related simplest-threshold function.Therefore we have to distribute the true-minterms of f_B in y_1 , y_2 , y_3 = 1 (or '0") spaces as (4 4 4).But the previous binary tree for f_G and f_D does not allow such a distribution see Figure 4.9.c .Hence we cannot transform the basic state function f_B to a simplest-threshold function.

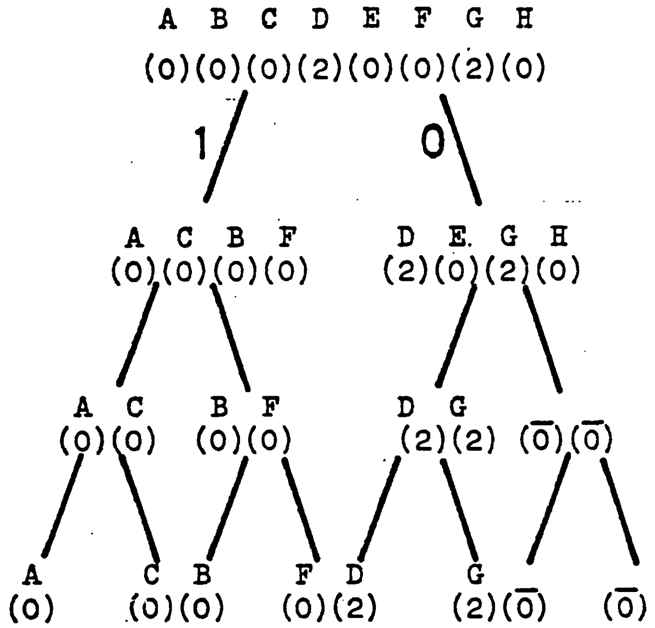
According to the list given at step 1 ,we will treat the basic state function f_C with four true-minterms.Again the first-order spectral coeffecients of the related simplest-threshold function are:

(8 8 8 0 0)

and the true-minterms distributions are:

(4 4 4 2 2).

The first-order absolute spectral values (8 0) of the basic state function f_C,which correspond to input variables, are among the first-order spectral coeffecients of the related simplest-threshold function. Therefore we have to distribute the true-minterms as (4 4 2)for the state variables y_1 , y_2 , y_3 . We must complete the binary tree in Figure 4.9.b.Since the basic state function f_B has failed to become a simplest-threshold function,the developed binary tree for the basic state function f_C is given in Figure 4.9.d.In order to satisfy the above mentioned distribution at the second partition for the y_2 variable , we have to give the same binary code to (A C) and (E -) sets, say " 1 " code.Again at the y_3 partition we have to give the same binary code to (C) and (E) sets to satisfy the four true-minterm distribution in y_3 = 1 (or '0') space for f_C.Now $f_C$is a simplest-



Basic state function f_B.

4 true-minterms in y_1=0 space.

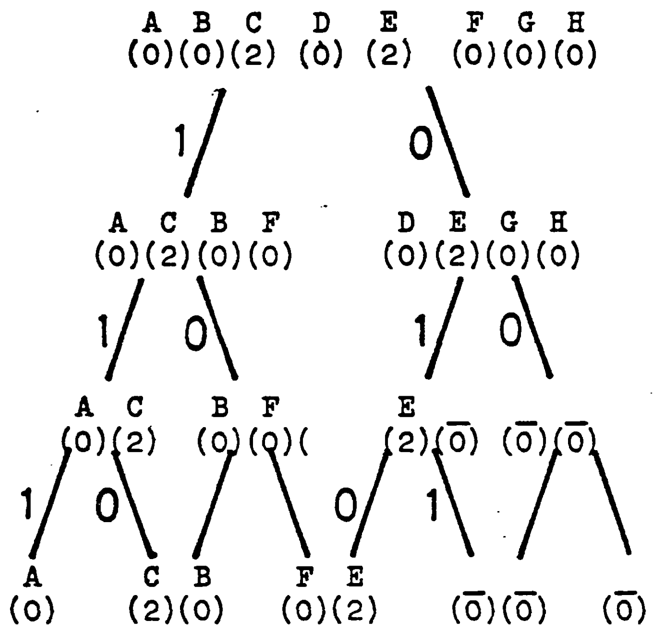
4 true-minterms in y_2=1(or y_2=0)space.

2 true-minterms in y_3=1 (or y_3=1)space.?

first-order spectral coeffecients of the related simplest-threshold δ : 8 8 8 0 0

and true.minterm distributions : (4) (4) (4) (2) (2)

Figure 4.9.c. Binary tree shows that basic state function f_B cannot be formed as simplest-threshold after f_G and f_D are treated.



Basic state function F_C

2 true-minterms in y_1=1 space.

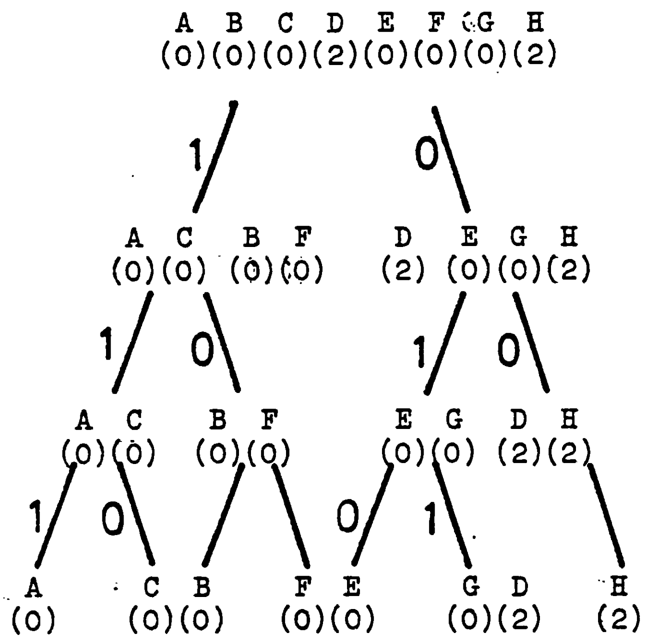
4 true-minterms in y_2=1 space.

4 true minterms in y_3=0 space.

first-order spectral coeffecients of the related simplest-threshold δ : 8 8 8 0 0

and true-minterm distributions: (4) (4) (4) (2) (2)

Figure 4.9.d Binary tree for the basic state functions f_C including f_G and f_D:



Basic state function F_E

4 true-minterms in
$y_1=1$ space.

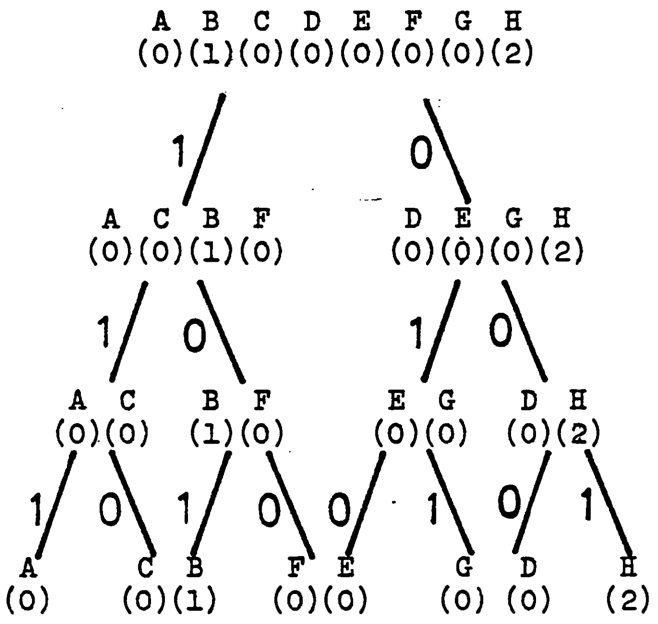
4 true-minterms in
$y_2=0$ space.

2 true minterms in
$y_3=1$(or $y_3=0$) space.

first-order spectral coeffecient of
the related-simplest-threshold δ : 8 8 8 0 0

and true-minterm distribution : 4 4 4 2 2

Figure 4.9.e. Binary tree developed for the basic state functions f_G, f_D and f_E.



Basic state function f_A.

2 true-minterms in
$y_1=0$ space.

3 true-minterms in
$y_2=0$ space.

3 true-minterms in
$y_3=1$ space.

first-order spectral coeffecients of
the related simplest-threshold δ : 6 6 6 2 2

and true-minterm distribution : 3 3 3 2 2

Figure 4.9.f Final binary tree developed for the basic state functions f_G, f_D, f_C, f_E and f_A.

threshold function because it has the same true-minterm distribution for y_1 , y_2 , y_3 as the related simplest-threshold function.

The basic state function f_E is the next one to be treated. The evaluation of the binary tree is continued in a similar way as in the previous step, see Figure 4.9.e.

The binary tree is not yet completed. We are able to continue with the basic state function f_A. Using the above procedure we obtain the final binary tree shown in Figure 4.9.f.

In this final binary tree, all the state codes have been determined as A (1 1 1) ; C (1 1 0) ; B (1 0 1) ; F (1 0 0) ; E (0 1 0) ; G (0 1 1) ; D (0 0 0) ; H (0 0 1). This means we cannot transform any other basic state function to a simplest-threshold function, because all the state codes have been determined. For this new state assignment the new state table and the next-state functions are given in Figure 4.10. Figure 4.11 shows the result state-table and next-state functions for Lala's method for the same example. The state codes for Lala's method is: A (0 0 1); B (1 0 1) ; C (1 0 0) ; D (1 1 1) ; E (0 1 1) ; F (0 0 0) ; G (1 1 0) ; H (0 1 0).

The resultant circuit diagrams for the new state assignment and the Lala's state assignment are shown in Figure 4.12 and Figure 4.13. Let us compare these circuits:

The new state assignment result:	Lala's state assignment result:
2 two-input AND gates	4 two-input AND gates
9 three-input AND gates	12 three-input AND gates
2 four-input AND gates	1 four-input AND gates
2 four-input OR gates	2 five-input OR gates
1 five-input OR gate	1 seven-input OR gate.
Total: 16 gates and 52 gate inputs	Total: 20 gates and 65 gate inputs

	$y_1y_2y_3$ \ x_1x_2	00	01	11	10
D	000	B	B	E	E
H	001	A	A	E	E
G	011	H	H	B	B
E	010	F	F	C	C
C	110	G	G	C	C
A	111	G	G	G	G
B	101	A	D	D	D
F	100	H	D	D	D

a) state table

$y_1y_2y_3$ \ x_1x_2	00	01	11	10
000	1	1		
001	1	1		
011			1	1
010	1	1	1	1
110			1	1
111				
101	1			
100				

b) $Y_1=\bar{x}_1\bar{y}_1\bar{y}_2+x_1\bar{y}_1y_2+\bar{y}_1y_2\bar{y}_3+x_1y_2\bar{y}_3+\bar{x}_1\bar{x}_2\bar{y}_2y_3$

$y_1y_2y_3$ \ x_1x_2	00	01	11	10
000			1	1
001	1	1	1	1
011				
010			1	1
110	1	1	1	1
111	1	1	1	1
101	1			
100				

c) $Y_2=y_1y_2+\bar{y}_1\bar{y}_2y_3+x_1\bar{y}_1\bar{y}_3+\bar{x}_1\bar{x}_2y_1y_3$

$y_1y_2y_3$ \ x_1x_2	00	01	11	10
000	1	1		
001	1	1		
011	1	1	1	1
010				
110	1	1		
111	1	1	1	1
101	1			
100	1			

d) $Y_3=y_2y_3+\bar{x}_1\bar{y}_1\bar{y}_2+\bar{x}_1y_1y_2+\bar{x}_1\bar{x}_2y_1$

Figure 4.10 The next-state functions for the new state assignment.

	$y_1y_2y_3$ \ x_1x_2	00	01	11	10
F	000	H	D	D	D
A	001	G	G	G	G
E	011	F	F	C	C
H	010	A	A	E	E
G	110	H	H	B	B
D	111	B	B	E	E
B	101	A	D	D	D
C	100	G	G	C	C

a) state table

$y_1y_2y_3$ \ x_1x_2	00	01	11	10
000		1	1	1
001	1	1	1	1
011			1	1
010				
110			1	1
111	1	1		
101		1	1	1
100	1	1	1	1

b) $Y_1'=y_1\bar{y}_2\bar{y}_3+\bar{y}_1\bar{y}_2y_3+x_1\bar{y}_2+x_2\bar{y}_2+x_1\bar{y}_1y_2+x_1y_1\bar{y}_3$
$+\bar{x}_1y_1y_2y_3$

$y_1y_2y_3$	00	01	11	10
000	1	1	1	1
001	1	1	1	1
011				
010			1	1
110	1	1		
111			1	1
101		1	1	1
100	1	1		

c) $Y_2'=\bar{y}_1\bar{y}_2+x_1\bar{y}_1\bar{y}_3+\bar{x}_1y_1\bar{y}_3+x_1y_1y_3$
$\bar{x}_1x_2\bar{y}_2$

$y_1y_2y_3$	00	01	11	10
000		1	1	1
001				
011				
010	1	1	1	1
110			1	1
111	1	1	1	1
101	1	1	1	1
100				

d) $Y_3'=y_1y_3+\bar{y}_1y_2\bar{y}_3+x_1\bar{y}_1\bar{y}_3+x_2\bar{y}_1\bar{y}_3+x_1y_2\bar{y}_3$

Figure 4.11 The next-state functions for Lala's state assignment.

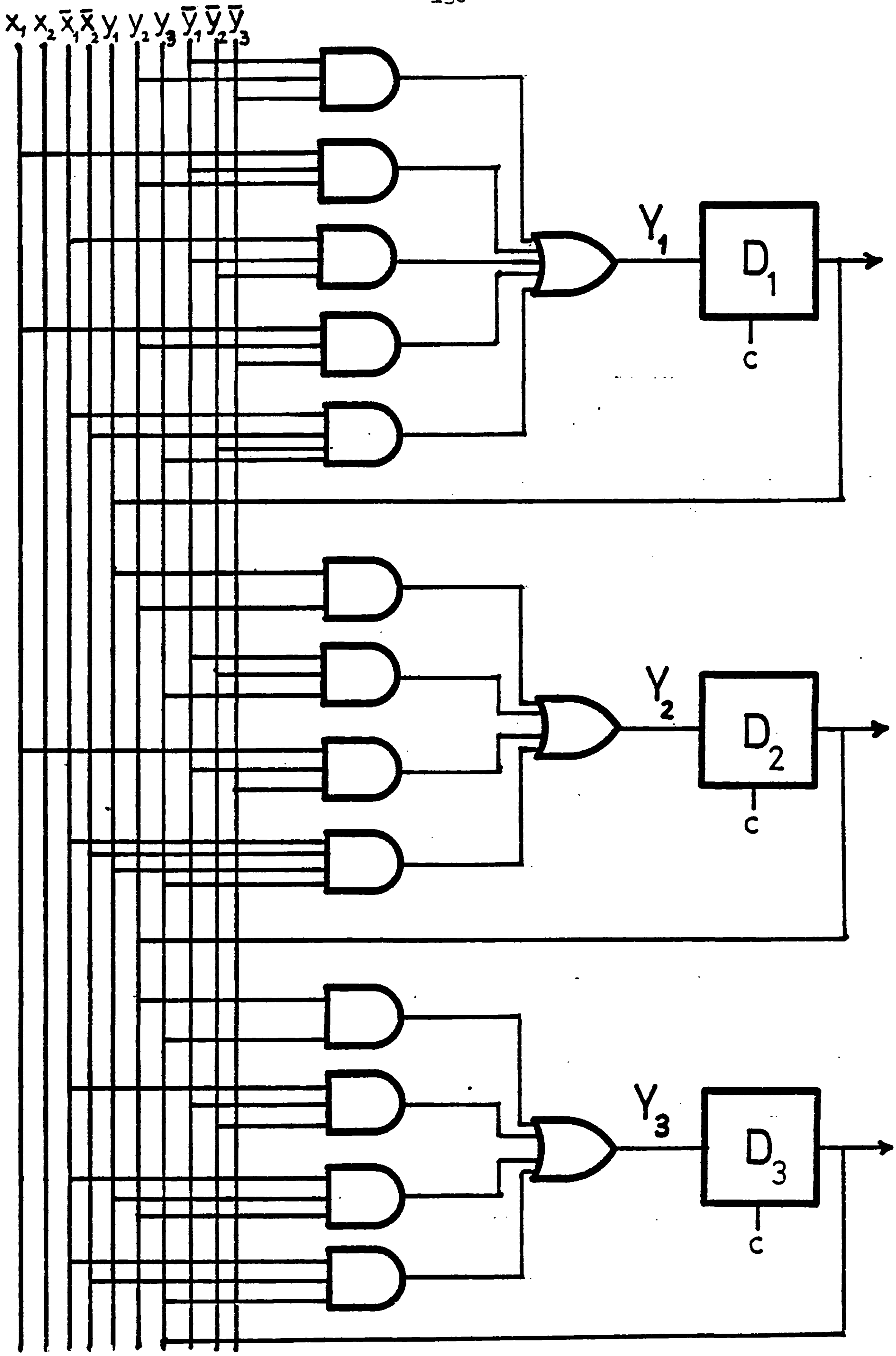


Figure 4.12 The circuit diagram of the example sequential machine in figure 4.8 when the new assignment technique applied.

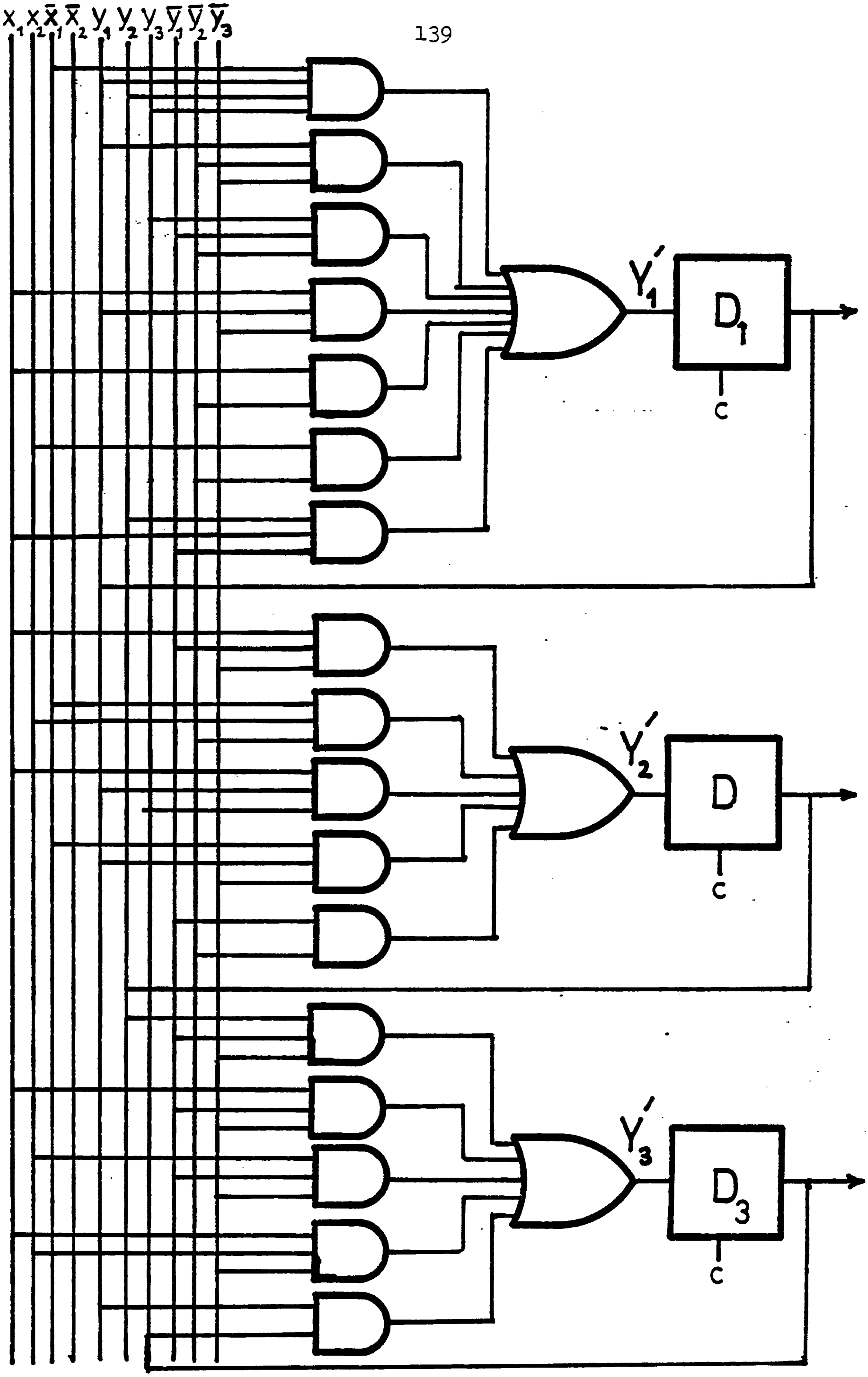


Figure 4.13 The circuit diagram of the sequential machine in Figure 4.8 when Lala's state assignment technique is applied.

4.4 Conclusion

This Chapter has been devoted to the application of minterm-interchange operation to the state assignment problem of sequential machine design.Altough this application is termed as "minterm - interchange" it becomes row interchange in the Karnaugh representation of basic state functions.However,row interchange is in fact a minterm-interchange of the corresponding minterms of the rows.

The basis of the given method for state assignment is to form the basic state functions as simplest-threshold functions, whereever possible. The reasons are:

i) next-state functions are disjoint combinations of basic state functions,

ii) simplest-threshold functions are simple and easy to determine.

The method was implemented by means of binary tree.

One of the advantages of employing basic state functions rather than directly using next-state functions is that realisations of the basic state functions,which are formed as simplest-threshold functions,can be used as a common circuit modules for the realisation of all next-state functions.This is because next-state functions or their complements are disjoint combinations of the basic state functions. More research is needed to investigate this potential advantage.

At the beginning of the procedure it was assumed that the sequential machine was complete. However this procedure can also be generilized to incomplete machines.Suppose that the next states in the rows of don't care states of an incomplete machine are chosen to be the states for which the number of appearances is highest in the state table,and the related basic state function can be formed as simplest-threshold functions, then in this case the method can be used to advantage for incomplete machines.

The binary tree explained in this Chapter is easy to handle by hand if the sequential machine has a small number of states. When the number of states increases the problem becomes cumbersome and is best handled by digital computer techniques.

This method does not always give the best solution to the state assignment problem. But it can be said that if the basic state functions are distributed in a few rows of the state table which represents the given sequential machine, then this method is useful. This is because such a property gives an opportunity to form many basic state functions as simplest-threshold functions.

REFERENCES

1. HILL, F.J. and PETERSON, G.R. " Introduction to Switching Theory and Logical Design" John-Wiley 1974 (second edition).

2. KOHAVI, Z. " Switching and Finite Automata Theory" McGraw-Hill., 1978

3. HARING, D.R. " Sequential Synthesis State Assignment Aspects" M.I.T. Research Monography No. 31. The M.I.T. Press Cambridge, Mass, 1966.

4. HARTMANIS, J. " On the State Assignment Problem for Sequential Machines I " IRE Trans. on Elect. Compt., p.p.157-165, June 1967.

5. HARTMANIS, J. and STEARNS, R.E. " On the State Assignment Problem for Sequential Machines II " IRE Trans. on Elect. Compt., p.p.593-603, December 1961.

6. CURTIS,H.A." Multiple Reduction of variable dependency of Sequential machines" J.ACM.,p.p.322-344 July 1962.

7. KARP,R.M. " Some Techniques of State Assignment for Synchronous Sequential Machines" IEEE Trans. Elect.Compt. Vol EC-13,p.p.507-518,October 1964.

8. KOHAVI,Z. " Secondary State Assignment for Sequential Machines" IEEE Trans. on Elect. Compt.,p.p.193-203,June 1964.

9. WEINER,P. and SMITH,E.J. " Optimization of Reduced Dependencies for Synchronous Machines" IEEE Trans. on Elect. Compt. Vol EC-16 p.p.835-847,December 1967.

10. HARLOW,C. and COATES,C.L. " On the Structure of Realisations Using Flip-flop Memory Elements" Information and Control 10, p.p.159-174,1967.

11. CURTIS,H.A. " Systematic Procedure for Realising Synchronous Sequential Machines Using Flip-flop Memory Part I " IEEE Trans. on Compt. Vol C-18 No 12,p.p.1121-1127,December 1969.

12. CURTIS,H.A. "Systematic Procedure for Realising Synchronous Sequential Machines Using Flip-flop Memory Part II " IEEE Trans. on Compt. Vol C-19 No 1,p.p.66-73,January 1970.

13. ARMSTRONG,D.B. " On the Efficient Assignment of Internal Codes to Sequential Machines" IRE Trans. on Elect. Compt.,p.p.611-622, October 1962.

14. DOLOTTA,T.A. and Mc.CLUSKEY,E.J." The Coding of Internal States of Sequential Circuits " IEEE Trans. on Elect.Compt.,p.p.549-562, October 1964.

15. TORNG,H.C. " An Algorithm for Finding Secondary Assignments of

Synchronous Sequential Circuits" IEEE Trans,on Compt. Vol C-17 No 5,p.p.461-469,May 1968.

16. STORY,J.R. ; HARRISON,H.J. and REINHARD,E.A. " Optimum State Assignment for Synchronous Sequential Circuits" IEEE Trans. on Compt. Vol C-21 No.12,p.p.1365-1373,December 1972.

17. LALA,P.K. " An Algorithm for the State Assignment of Synchronous Sequential Circuits" Electronics Letters Vol 14, No.6,p.p.199-201,March 1978.

18. EDWARDS,C.R. and ERIS,E. " State Assignment and Entropy" Electronics Letters Vol 14,No.13,p.p.390-391, 22 June 1978.

CHAPTER 5 GENERAL CONCLUSIONS

GENERAL CONCLUSIONS

As is indicated by the title of this thesis,the given combinational circuit design and state assignment techniques are developed from the basic operation of " minterm-interchange ".

The circuit implementation of this operation is an exclusive-OR of the functions δ and γ That is

$$f = \delta \oplus \gamma$$

δ is determined by minterm-interchange on the Boolean function f,and γ is an auxiliary function to form δ from f. Clearly both δ and f have the same number of true-minterms,since one of them is obtained from the other by minterm-interchange.

First the minterm-interchange application for combinational networks was considered.It was found desirable to have the δ and γ functions as simple functions in the exclusive-OR decomposition of f. To this end the "complexity concept" was required to decide the δ and γ functions.

Since δ has the same number of true-minterms as the original function f, it is necessary to define "simplest functions" for a given number of true-minterms and they must also be easy to recognize.Thus δ functions were chosen as "simplest-threshold functions". These functions are simple to design and also can be characterised by n+1 coefficients known as the Chow parameters.Because it is easy to manipulate this particular type of threshold function in the spectral domain , the minterm-interchange operation was examined in this domain.

The class of simplest-threshold functions were extensively used.This does not necessarily require the use of threshold gates in practical realisations.Hewover if threshold gates become commercially available this method will show advantage *.

One of the most advantageous properties of simplest-threshold functions is that they can be used as modules in practical circuits, since once the order and the number of true-minterms are known,then the corresponding simplest-threshold function can be defined and realised (excepting additional NOT gates).

The number of true-minterms of the γ function depends on the number of interchanged-minterms in the function f. Thence in order to determine the complexity of the γ function, "entropy complexity" has been exploited because the entropy complexity of a Boolean function is based on the number of true-minterms in the function.

As a result of combining the concepts explained above,an algorithm was given for the combinational circuit design by minterm-interchange.

It should be noted that the new method is an alternative to the conventional combinational circuit design techniques.The advantages of this method,when compared to conventional approaches,are limited by:

a) the conventional realisation cost of the Boolean function to be designed,

b) cost of the γ function compared with the cost of the Boolean

---

* The difficulties of industrially fabricating threshold logic gates with good operating tolerances is well known [1],and has to date precluded their introduction on the commercial market.However they represent a class of functions which if made would provide a more powerful logical use of silicon area than conventional vertex(AND,OR,NAND,NOR)logic gates.

function f.

There remain two open problems to be solved in this area:

i) The technique was developed for only complete Boolean functions, and does not cover Don't Care functions.This is both an open problem and a difficult one,because the choice of don't care outputs,as 0 or 1 , changes the number of true-minterms in the original function whereas the number of true-minterms remains invariant under minterm-interchange operation.

ii)The method of minterm-interchange can also be used in the design of multi-output networks. This is because two functions with the same number of true-minterms may be generated from each other by the minterm-interchange operation.Hence it is possible to design functions with the same number of true-minterms by applying the minterm-interchange operation. More research is needed in this area.

The second application of minterm-interchange considered in this thesis is that of state assignment for synchronous and complete sequential machines.

It is known that if delay type memory elements are used,the binary codes given to the states are normally chosen such that the next-state functions become simple.This minimises the combinational part of the circuitry of the sequential machine.

We cannot directly simplify next-state functions by using minterm-interchange (which correspond to a state assignment change). However , "basic state functions" can be formed as simplest-threshold by minterm interchange.Further the next-state functions are the AND or exclusive-OR combinations of basic state functions.Thus the state assignment method developed in this thesis was based upon forming the

the basic state functions as simplest-threshold.

In Chapter 4,an algorithm was given which employs a binary tree.Like other methods,this new technique cannot guarantee a best solution to the state assignment problem.It is simply a valuable alternative approach to perhaps the most difficult problem of the sequential machine design.

Some of the topics not considered in this thesis are:

1) generalisation of the method to incomplete sequential machines
2) application of the technique to the design of sequential machines with J-K (or S-R) type memory elements instead of delay type
3) compilation of a computer programme for optimal state assignment in large sequential machines
4) applying the method to use the basic state functions (transformed to simplest-thresholds) as integrated circuit modules
5) the problem of testing digital networks.This problem has an increasing practical importance particularly in the sequential network area.

Reference

1. HURST,S.L. "The Logical Processing of Digital Signals" Edward Arnold,London,Appendix E,1978.

APPENDIX A ; SIMPLEST AND SIMPLEST-THRESHOLD FUNCTIONS

In this Appendix simplest functions are given for $n \leqslant 5$. These functions are determined when the n:number of variables and u:number of true-minterms are known.The simplest functions for a given u and $u' = 2^n - u$ are the same,because the cost of the additional inversion inversion which distinguishes them has not been taken into account.In the look-up table,the Karnaugh map representation of simplest functions and also the first-order spectral coeffecients are given.

The first-order coeffecients are in positive canonic form. Then all of the simplest functions which are in the same NP class can be found by NP manipulations.These first-order spectral coeffecients are necessary and sufficient to decide that the simplest function is threshold or not by looking at the list in Appendix B.

The values in the second line under the first-order spectral coeffecients show the true-minterm distribution in the spaces where the independent variables take the value 1 (see Appendix D for the spaces). The figure below illustrates the meaning of the numbers given in the look-up table.

	first-order spectral coefficients		
R_0	R_1	R_2	R_3
4	4	4	0
(2)	(2)	(2)	(1)

u:number of true-minterms in the function → (2)

(2): number of true minterms in the space $x_1=1$

(2): number of true minterm in the space $x_2=1$

(1): number of true minterms in the space $x_3=1$

We will illustrate how the true-minterm distributions of simplest-threshold functions have been calculated,It is known from equation 1.24 that the relationship between true-minterm distribution and first-order spectral eoeffecients is as follows :

$$Z_i = \frac{1}{4} (R_i+2u)$$

where

Z_i: Number of true-minterms in x_i=1 space,

R_i: First-order spectral coeffecient corresponding to the independent variable x_i,

u : total true-minterms of the function.

The additional relationship between R_o and number of true-minterms is given by:

$$2u = 2^n-R_o$$

Now let us calculate the true-minterm distribution for n=4,u=5 simplest-threshold function.Spectral coeffecients are 6 10 6 2 2 in the order R_o R_1 R_2 R_3 R_4 .

Number of true-minterms in the function:

$$u = \frac{2^n-R_o}{2} = \frac{16-6}{2} = 5$$

Number of true-minterms in x_1=1 space:

$$Z_1= \frac{1}{4} (R_1+2u) = \frac{1}{4} (10+2\times5)=5$$

Number of true-minterms in x_2=1 space:

$$Z_2= \frac{1}{4} (R_2+2u) = \frac{1}{4} (6+2\times5)=4$$

Number of true-minterms in x_3=1 space:

$$Z_3= \frac{1}{4}(R_3+2u)= \frac{1}{4}(2+2\times5)=3$$

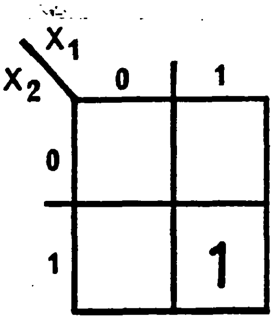
Number of true-minterms in x_4=1 space:

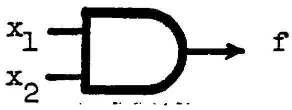
$$Z_4= \frac{1}{4}(R_4+2u)= \frac{1}{4}(2+2\times5)=3$$

These minterm distributions are shown in parenthesis under the corresponding first-order spectral coefficients in the look-up tables. Note that given Z values are for the small 'u' values.

n = 2

$\underline{n = 2}$, $\underline{u = 1, 3}$



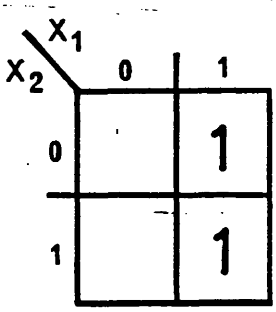


$f = x_1 x_2$,

THRESHOLD ·

R_o	R_1	R_2
2	2	2
(1)	(1)	(1)

$\underline{n = 2}$, $\underline{u = 2}$



$f = x_1$,

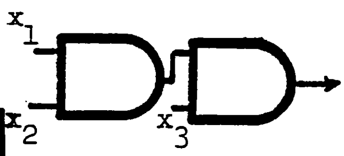
THRESHOLD .

R_0	R_1	R_2
0	4	0
(2)	(2)	(1)

n=3

$\underline{n = 3}$, $\underline{u = 1,7}$

x_3 \ x_1x_2	00	01	11	10
0				
1			1	



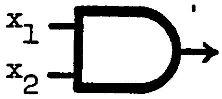
$f = x_1x_2x_3$,

THRESHOLD.

	R_0	R_1	R_2	R_3
	6	2	2	2
u =	(1)	(1)	(1)	(1)

$\underline{n = 3}$, $\underline{u = 2,6}$

x_3 \ x_1x_2	00	01	11	10
0			1	
1			1	



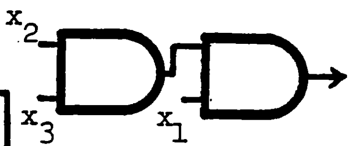
$f = x_1x_2$,

THRESHOLD.

	R_0	R_1	R_2	R_3
	4	4	4	0
u =	(2)	(2)	(2)	(1)

$\underline{n = 3}$, $\underline{u = 3,5}$

x_3 \ x_1x_2	00	01	11	10
0			1	
1			1	1



$f = x_1(x_2+x_3)$,

THRESHOLD.

	R_0	R_1	R_2	R_3
	2	6	2	2
u =	(3)	(3)	(2)	(2)

$\underline{n = 3}$, $\underline{u = 4}$

x_3 \ x_1x_2	00	01	11	10
0			1	1
1			1	1

$f = x_1$,

THRESHOLD.

	R_0	R_1	R_2	R_3
	0	8	0	0
u =	(4)	(4)	(2)	(2)

n = 4

$n = 4$,

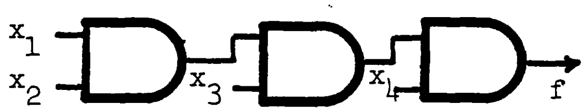
$u = 1, 15$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00				
01				
11			1	
10				

$f = x_1x_2x_3x_4$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
14	2	2	2	2
(1)	(1)	(1)	(1)	(1)



$n = 4$,

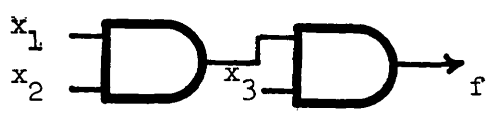
$u = 2 , 14$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00				
01				
11			1	
10			1	

$f = x_1x_2x_3$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
12	4	4	4	0
(2)	(2)	(2)	(2)	(1)



$n = 4$,

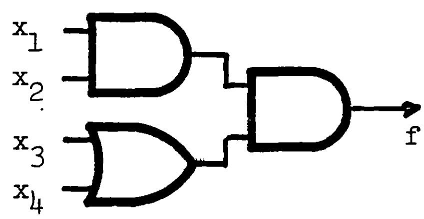
$u = 3 , 13$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00				
01			1	
11			1	
10			1	

$f = x_1x_2 (x_3 + x_4)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
10	6	6	2	2
(3)	(3)	(3)	(2)	(2)



$n = 4$ continued

$n = 4$,

$u = 4$, 12

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	
01			1	
11			1	
10			1	

$f = x_1x_2$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
8	8	8	0	0
(4)	(4)	(4)	(2)	(2)

$n = 4$,

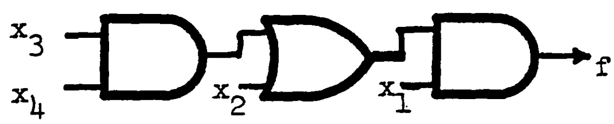
$u = 5$, 11

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	
01			1	
11			1	1
10			1	

$f = x_1 (x_2 + x_3x_4)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
6	10	6	2	2
(5)	(5)	(4)	(3)	(3)



$n = 4$,

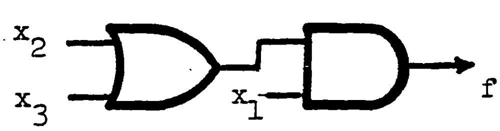
$u = 6$, 10

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	
01			1	
11			1	1
10			1	1

$f = x_1 (x_2 + x_3)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
4	12	4	4	0
(6)	(6)	(4)	(4)	(3)



$n = 4$ continued

$n = 4$,

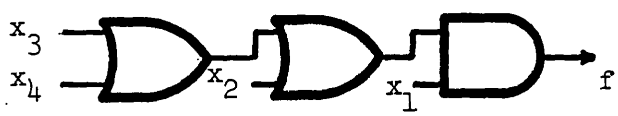
$u = 7$, 9

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	
01			1	1
11			1	1
10			1	1

$f = x_1(x_2+x_3+x_4)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
2	14	2	2	2
(7)	(7)	(4)	(4)	(4)



$n = 4$,

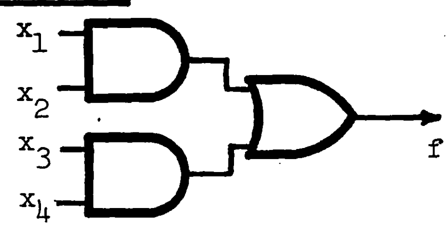
$u = 7$, 9

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	
01			1	
11	1	1	1	1
10			1	

$f = x_1x_2 + x_3x_4$,

NONTHRESHOLD .

R_0	R_1	R_2	R_3	R_4
2	6	6	6	6
(7)	(5)	(5)	(5)	(5)



$n = 4$,

$u = 8$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00			1	1
01			1	1
11			1	1
10			1	1

$f = x_1$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4
0	16	0	0	0
(8)	(8)	(4)	(4)	(4)

n = 5

n = 5 ,

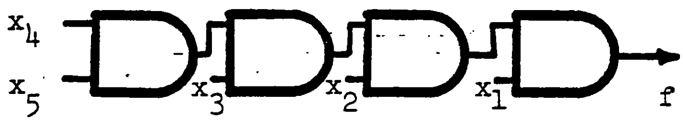
u = 1 , 31

x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
	$x_1 = 0$				$x_1 = 1$			
00								
01								
11						1		
10								

$f = x_1x_2x_3x_4x_5$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
30	2	2	2	2	2
(1)	(1)	(1)	(1)	(1)	(1)



n = 5 ,

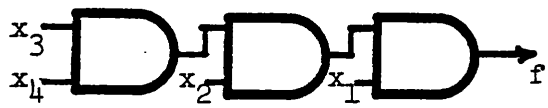
u = 2 , 30

x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
	$x_1 = 0$				$x_1 = 1$			
00								
01								
11						1		
10						1		

$f = X_1x_2x_3x_4$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
28	4	4	4	4	0
(2)	(2)	(2)	(2)	(2)	(1)



n = 5 ,

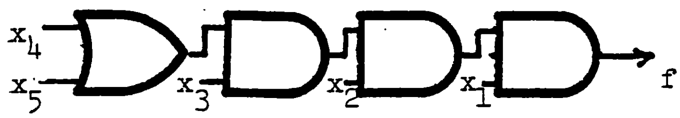
u = 3 , 29

x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
	$x_1 = 0$				$x_1 = 1$			
00								
01						1		
11						1		
10						1		

$f = x_1x_2x_3(x_4+x_5)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
26	6	6	6	2	2
(3)	(3)	(3)	(3)	(2)	(2)



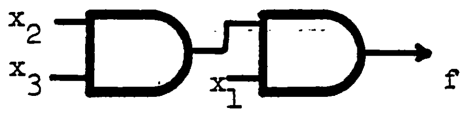
n=5 continued

$n = 5$, $u = 4$, 28

$x_4x_5 \backslash x_2x_3$	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00						1		
01						1		
11						1		
10						1		

$f = x_1x_2x_3$, THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
24	8	8	8	0	0
(4)	(4)	(4)	(4)	(2)	(2)

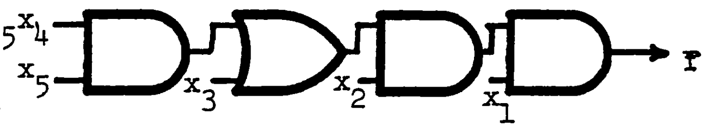


$n = 5$, $u = 5$, 27

$x_4x_5 \backslash x_2x_3$	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00						1		
01						1		
11					1	1		
10						1		

$f=x_1x_2(x_3+x_4x_5)$, THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
22	10	10	6	2	2
(5)	(5)	(5)	(4)	(3)	(3)

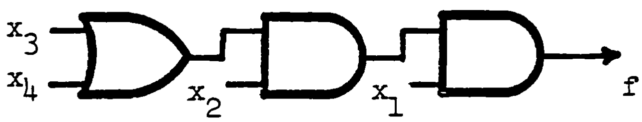


$n = 5$, $u = 6$, 26

$x_4x_5 \backslash x_2x_3$	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00						1		
01						1		
11					1	1		
10					1	1		

$f=x_1x_2(x_3+x_4)$, THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
20	12	12	4	4	0
(6)	(6)	(6)	(4)	(4)	(3)



n = 5 continued

	$x_1 = 0$				$x_1 = 1$			
x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
00						1		
01					1	1		
11					1	1		
10					1	1		

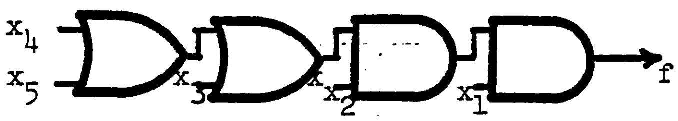
$n = 5$,

$u = 7$, 25

$f = x_1 x_2 (x_3 + x_4 + x_5)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
18	14	14	2	2	2
(7)	(7)	(7)	(4)	(4)	(4)



	$x_1 = 0$				$x_1 = 1$			
x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
00						1		
01						1		
11					1	1	1	1
10						1		

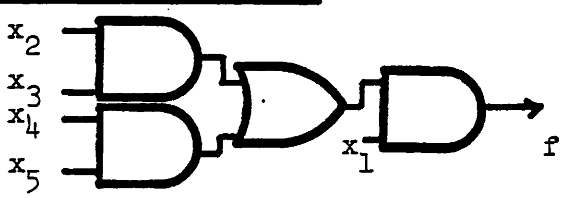
$n = 5$,

$u = 7$,25

$f = x_1 (x_2 x_3 + x_4 x_5)$,

NONTHRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
18	14	6	6	6	6
(7)	(7)	(5)	(5)	(5)	(5)



	$x_1 = 0$				$x_1 = 1$			
x_4x_5 \ x_2x_3	00	01	11	10	10	11	01	00
00					1	1		
01					1	1		
11					1	1		
10					1	1		

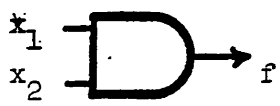
$n = 5$,

$u = 8$, 24

$f = x_1 x_2$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
16	16	16	0	0	0
(8)	(8)	(8)	(4)	(4)	(4)



n = 5 continued

$x_4x_5 \backslash x_2x_3$	$x_1 = 0$				$x_1 = 1$			
	00	01	11	10	10	11	01	00
00					1	1		
01					1	1		
11					1	1	1	
10					1	1		

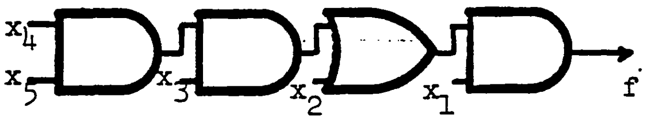
$n = 5$,

$u = 9$, 23

$f = x_1(x_2 + x_3x_4x_5)$,

THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
14	18	14	2	2	2
(9)	(9)	(8)	(4)	(4)	(4)



$x_4x_5 \backslash x_2x_3$	$x_1 = 0$				$x_1 = 1$			
	00	01	11	10	10	11	01	00
00					1	1		
01					1	1		
11					1	1	1	
10					1	1	1	

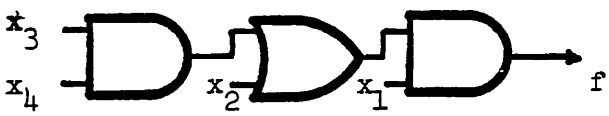
$n = 5$,

$u = 10$, 22

$f = x_1(x_2 + x_3x_4)$,

THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
12	20	12	4	4	0
(10)	(10)	(8)	(6)	(6)	(5)



$x_4x_5 \backslash x_2x_3$	$x_1 = 0$				$x_1 = 1$			
	00	01	11	10	10	11	01	00
00					1	1		
01					1	1	1	
11					1	1	1	
10					1	1	1	

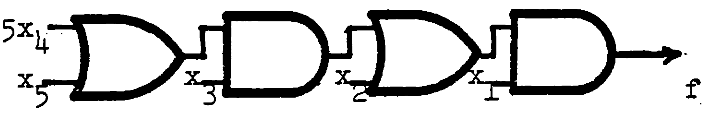
$n = 5$,

$u = 11$, 21

$f = x_1\left[x_2 + x_3(x_4 + x_5)\right]$,

THRESHOLD.

R_0	R_1	R_2	R_3	R_4	R_5
10	22	10	6	2	2
(11)	(11)	(8)	(7)	(6)	(6)



n = 5

cntinued

$n = 5$,

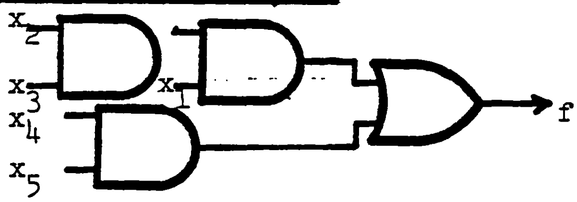
$u = 11 , 21$

$x_4x_5 \backslash x_2x_3$	00 ($x_1 = 0$)	01	11	10	10 ($x_1 = 1$)	11	01	00
00						1		
01						1		
11	1	1	1	1	1	1	1	1
10						1		

$f = x_1x_2x_3 + x_4x_5$,

NONTHRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
10	6 .	6	6	14	14
(11)	(7)	(7)	(7)	(9)	(9)



$n = 5$,

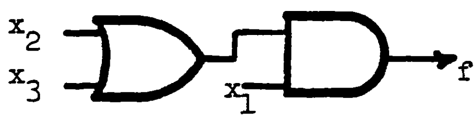
$u = 12 , 20$

$x_4x_5 \backslash x_2x_3$	00 ($x_1 = 0$)	01	11	10	10 ($x_1 = 1$)	11	01	00
00					1	1	1	
01					1	1	1	
11					1	1	1	
10					1	1	1	

$f = x_1(x_2 + x_3)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
8	24	8	8	0	0
(12)	(12)	(8)	(8)	(6)	(6)



$n = 5$,

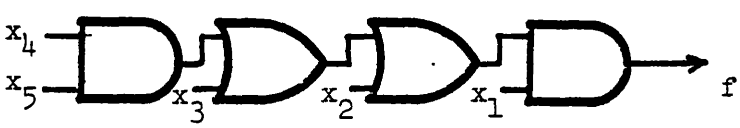
$U = 13 , 19$

$x_4x_5 \backslash x_2x_3$	00 ($x_1 = 0$)	01	11	10	10 ($x_1 = 1$)	11	01	00
00					1	1	1	
01					1	1	1	
11					1	1	1	1
10					1	1	1	

$f = x_1(x_2 + x_3 + x_4x_5)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
6	26	6	6	2	2
(13)	(13)	(8)	(8)	(7)	(7)



n = 5

continued

n = 5 ,

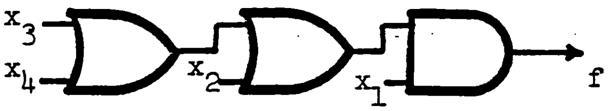
u = 14 , 18

x_4x_5 \ x_2x_3	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00					1	1		
01					1	1	1	1
11					1	1	1	1
10					1	1	1	1

$f=x_1(x_2+x_3+x_4)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
4	28	4	4	4	0
(14)	(14)	(8)	(8)	(8)	(7)



n = 5 ,

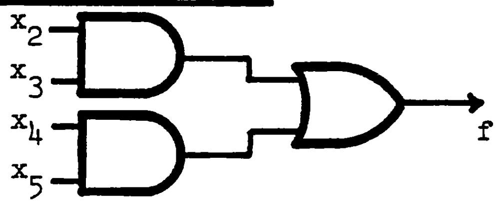
u = 14 , 18

x_4x_5 \ x_2x_3	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00			1			1		
01			1			1		
11	1	1	1	1	1	1	1	1
10			1			1		

$f=x_2x_3+x_4x_5$,

NONTHRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
4	0	12	12	12	12
(14)	(7)	(10)	(10)	(10)	(10)



n = 5 ,

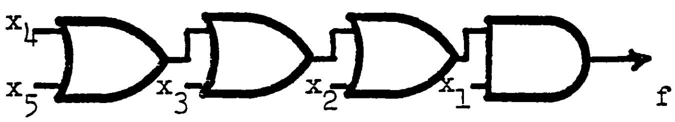
u = 15 , 17

x_4x_5 \ x_2x_3	$x_1=0$: 00	01	11	10	$x_1=1$: 10	11	01	00
00					1	1	1	
01					1	1	1	1
11					1	1	1	1
10					1	1	1	1

$f=x_1(x_2+x_3+x_4x_5)$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
2	30	2	2	2	2
(15)	(15)	(8)	(8)	(8)	(8)



n = 5

continued

n = 5 ,

u = 16

$x_4x_5 \backslash x_2x_3$	$x_1 = 0$: 00	01	11	10	$x_1 = 1$: 10	11	01	00
00					1	1	1	1
01					1	1	1	1
11					1	1	1	1
10					1	1	1	1

$f = x_1$,

THRESHOLD .

R_0	R_1	R_2	R_3	R_4	R_5
0	32	0	0	0	0
(16)	(16)	(8)	(8)	(8)	(8)

APPENDIX B

CANONIC CHARACTERISTIC WEIGHT-THRESHOLD VECTORS, or CHOW PARAMETERS, FOR THRESHOLD FUNCTIONS OF UP TO n = 6.

n(max)	No.	c]						w]					
3													
	1	8	0	0	0			1	0	0	0		
	2	6	2	2	2			2	1	1	1		
	3	4	4	4	0			1	1	1	0		
4													
	1	16	0	0	0	0		1	0	0	0	0	
	2	14	2	2	2	2		3	1	1	1	1	
	3	12	4	4	4	0		2	1	1	1	0	
	4	10	6	6	2	2		3	2	2	1	1	
	5	8	8	8	0	0		1	1	1	0	0	
	6	8	8	4	4	4		2	2	1	1	1	
	7	6	6	6	6	6		1	1	1	1	1	
5													
	1	32	0	0	0	0	0	1	0	0	0	0	0
	2	30	2	2	2	2	2	4	1	1	1	1	1
	3	28	4	4	4	4	0	3	1	1	1	1	0
	4	26	6	6	6	2	2	5	2	2	2	1	1
	5	24	8	8	4	4	4	4	2	2	1	1	1
	6	24	8	8	8	0	0	2	1	1	1	0	0
	7	22	10	10	6	2	2	5	3	3	2	1	1
	8	22	10	6	6	6	6	3	2	1	1	1	1
	9	20	12	12	4	4	0	3	2	2	1	1	0
	10	20	12	8	8	4	4	4	3	2	2	1	1
	11	20	8	8	8	8	8	2	1	1	1	1	1
	12	18	14	14	2	2	2	4	3	3	1	1	1
	13	18	14	10	6	6	2	5	4	3	2	2	1
	14	18	10	10	10	6	6	3	2	2	2	1	1
	15	16	16	16	0	0	0	1	1	1	0	0	0
	16	16	16	12	4	4	4	3	3	2	1	1	1
	17	16	16	8	8	8	0	2	2	1	1	1	0
	18	16	12	12	8	8	4	4	3	3	2	2	1
	19	14	14	14	6	6	6	2	2	2	1	1	1
	20	14	14	10	10	10	2	3	3	2	2	2	1
	21	12	12	12	12	12	0	1	1	1	1	1	0

$n \leq 6$

1	64	0	0	0	0	0	0	1	0	0	0	0	0	0
2	62	2	2	2	2	2	2	5	1	1	1	1	1	1
3	60	4	4	4	4	4	0	4	1	1	1	1	1	0
4	58	6	6	6	6	2	2	7	2	2	2	2	1	1
5	56	8	8	8	8	0	0	3	1	1	1	1	0	0
6	56	8	8	8	4	4	4	6	2	2	2	1	1	1
7	54	10	10	10	6	2	2	8	3	3	3	2	1	1
8	54	10	10	6	6	6	6	5	2	2	1	1	1	1
9	52	12	12	12	4	4	0	5	2	2	2	1	1	0
10	52	12	12	8	8	4	4	7	3	3	2	2	1	1
11	52	12	8	8	8	8	8	4	2	1	1	1	1	1
12	50	14	14	14	2	2	2	7	3	3	3	1	1	1
13	50	14	14	10	6	6	2	9	4	4	3	2	2	1
14	50	14	10	10	10	6	6	6	3	2	2	2	1	1
15	50	10	10	10	10	10	10	3	1	1	1	1	1	1
16	48	16	16	16	0	0	0	2	1	1	1	0	0	0
17	48	16	16	12	4	4	4	6	3	3	2	1	1	1
18	48	16	16	8	8	8	0	4	2	2	1	1	1	0
19	48	16	12	12	8	8	4	8	4	3	3	2	2	1
20	48	12	12	12	12	8	8	5	2	2	2	2	1	1
21	46	18	18	14	2	2	2	7	4	4	3	1	1	1
22	46	18	18	10	6	6	2	9	5	5	3	2	2	1
23	46	18	14	14	6	6	6	5	3	2	2	1	1	1
24	46	18	14	10	10	10	2	7	4	3	2	2	2	1
25	46	14	14	14	10	10	6	7	3	3	3	2	2	1
26	44	20	20	12	4	4	0	5	3	3	2	1	1	0
27	44	20	20	8	8	4	4	7	4	4	2	2	1	1
28	44	20	16	16	4	4	4	6	4	3	3	1	1	1
29	44	20	16	12	8	8	4	8	5	4	3	2	2	1
30	44	20	12	12	12	12	0	3	2	1	1	1	1	0
31	44	16	16	16	8	8	8	4	2	2	2	1	1	1
32	44	16	16	12	12	12	4	6	3	3	2	2	2	1
33	42	22	22	10	6	2	2	8	5	5	3	2	1	1
34	42	22	22	6	6	6	6	5	3	3	1	1	1	1
35	42	22	18	14	6	6	2	9	6	5	4	2	2	1
36	42	22	18	10	10	6	6	6	4	3	2	2	1	1
37	42	22	14	14	10	10	2	7	5	3	3	2	2	1
38	42	18	18	18	6	6	6	5	3	3	3	1	1	1
39	42	18	18	14	10	10	6	7	4	4	3	2	2	1
40	42	18	14	14	14	14	2	5	3	2	2	2	2	1
41	40	24	24	8	8	0	0	3	2	2	1	1	0	0
42	40	24	24	8	4	4	4	6	4	4	2	1	1	1
43	40	24	20	12	8	4	4	7	5	4	3	2	1	1
44	40	24	20	8	8	8	8	4	3	2	1	1	1	1
45	40	24	16	16	8	8	0	4	3	2	2	1	1	0
46	40	24	16	12	12	8	4	8	6	4	3	3	2	1
47	40	20	20	16	8	8	4	8	5	5	4	2	2	1
48	40	20	20	12	12	8	8	5	3	3	2	2	1	1
49	40	20	16	16	12	12	4	6	4	3	3	2	2	1
50	40	16	16	16	16	16	0	2	1	1	1	1	1	0
51	38	26	26	6	6	2	2	7	5	5	2	2	1	1
52	38	26	22	10	10	2	2	8	6	5	3	3	1	1
53	38	26	22	10	6	6	6	5	4	3	2	1	1	1
54	38	26	18	14	10	6	2	9	7	5	4	3	2	1
55	38	26	18	10	10	10	6	6	5	3	2	2	2	1
56	38	26	14	14	14	6	6	5	4	2	2	2	1	1
57	38	22	22	14	10	6	6	6	4	4	3	2	1	1

58	38	22	22	10	10	10	10	3	2	2	1	1	1	1
59	38	22	18	18	10	10	2	7	5	4	4	2	2	1
60	38	22	18	14	14	10	6	7	5	4	3	3	2	1
61	38	18	18	18	14	14	2	5	3	3	3	2	2	1
62	36	28	28	4	4	4	0	4	3	3	1	1	1	0
63	36	28	24	8	8	4	4	6	5	4	2	2	1	1
64	36	28	20	12	12	4	0	5	4	3	2	2	1	0
65	36	28	20	12	8	8	4	7	6	4	3	2	2	1
66	36	28	16	16	12	4	4	6	5	3	3	2	1	1
67	36	28	16	12	12	8	8	8	7	4	3	3	2	2
68	36	24	24	12	12	4	4	7	5	5	3	3	1	1
69	36	24	24	12	8	8	8	4	3	3	2	1	1	1
70	36	24	20	16	12	8	4	8	6	5	4	3	2	1
71	36	24	20	12	12	12	8	5	4	3	2	2	2	1
72	36	24	16	16	16	8	8	4	3	2	2	2	1	1
73	36	20	20	20	12	12	0	3	2	2	2	1	1	0
74	36	20	20	16	16	12	4	6	4	4	3	3	2	1
75	34	30	30	2	2	2	2	5	4	4	1	1	1	1
76	34	30	26	6	6	6	2	7	6	5	2	2	2	1
77	34	30	22	10	10	6	2	8	7	5	3	3	2	1
78	34	30	18	14	14	2	2	7	6	4	3	3	1	1
79	34	30	18	14	10	6	6	9	8	5	4	3	2	2
80	34	30	14	14	10	10	10	7	6	3	3	2	2	2
81	34	26	26	10	10	6	6	5	4	4	2	2	1	1
82	34	26	22	14	14	6	2	9	7	6	4	4	2	1
83	34	26	22	14	10	10	6	6	5	4	3	2	2	1
84	34	26	18	18	14	6	6	5	4	3	3	2	1	1
85	34	26	18	14	14	10	10	6	5	4	3	3	2	2
86	34	22	22	18	14	10	2	7	5	5	4	3	2	1
87	34	22	22	14	14	14	6	4	3	3	2	2	2	1
88	34	22	18	18	18	10	6	5	4	3	3	3	2	1
89	32	32	32	0	0	0	0	1	1	1	0	0	0	0
90	32	32	28	4	4	4	4	4	4	3	1	1	1	1
91	32	32	24	8	8	8	0	3	3	2	1	1	1	0
92	32	32	20	12	12	4	4	5	5	3	2	2	1	1
93	32	32	16	16	16	0	0	2	2	1	1	1	0	0
94	32	32	16	16	8	8	8	4	4	2	2	1	1	1
95	32	32	12	12	12	12	12	3	3	1	1	1	1	1
96	32	28	28	8	8	8	4	6	5	5	2	2	2	1
97	32	28	24	12	12	8	4	7	6	5	3	3	2	1
98	32	28	20	16	16	4	4	6	5	4	3	3	1	1
99	32	28	20	16	12	8	8	7	6	5	4	3	2	2
100	32	28	16	16	12	12	12	5	4	3	3	2	2	2
101	32	24	24	16	16	8	0	4	3	3	2	2	1	0
102	32	24	24	16	12	12	4	5	4	4	3	2	2	1
103	32	24	20	20	16	8	4	6	5	4	4	3	2	1
104	32	24	20	16	16	12	8	7	6	5	4	4	3	2
105	32	20	20	20	20	8	8	3	2	2	2	2	1	1
106	30	30	30	6	6	6	6	3	3	3	1	1	1	1
107	30	30	26	10	10	10	2	5	5	4	2	2	2	1
108	30	30	22	14	14	6	6	4	4	3	2	2	1	1
109	30	30	18	18	18	2	2	5	5	3	3	3	1	1
110	30	30	18	18	10	10	10	3	3	2	2	1	1	1
111	30	30	14	14	14	14	14	2	2	1	1	1	1	1
112	30	26	26	14	14	10	2	6	5	5	3	3	2	1
113	30	26	22	18	18	6	2	7	6	5	4	4	2	1
114	30	26	22	18	14	10	6	8	7	6	5	4	3	2
115	30	26	18	18	14	14	10	6	5	4	4	3	3	2
116	30	22	22	22	18	6	6	4	3	3	3	2	1	1

117	30	22	22	18	18	10	10	5	4	4	3	3	2	2
118	28	28	28	12	12	12	0	2	2	2	1	1	1	0
119	28	28	24	16	16	8	4	5	5	4	3	3	2	1
120	28	28	20	20	20	4	0	3	3	2	2	2	1	0
121	28	28	20	20	12	12	8	4	4	3	3	2	2	1
122	28	28	16	16	16	16	12	3	3	2	2	2	2	1
123	28	24	24	20	20	4	4	5	4	4	3	3	1	1
124	28	24	24	20	16	8	8	6	5	5	4	3	2	2
125	28	24	20	20	16	12	12	7	6	5	5	4	3	3
126	26	26	26	18	18	6	6	3	3	3	2	2	1	1
127	26	26	22	22	22	2	2	4	4	3	3	3	1	1
128	26	26	22	22	14	10	10	5	5	4	4	3	2	2
129	26	26	18	18	18	14	14	4	4	3	3	3	2	2
130	26	22	22	22	14	14	14	4	3	3	3	2	2	2
131	24	24	24	24	24	0	0	1	1	1	1	1	0	0
132	24	24	24	24	12	12	12	2	2	2	2	1	1	1
133	24	24	20	20	16	16	16	5	5	4	4	3	3	3
134	22	22	22	18	18	18	18	3	3	3	2	2	2	2
135	20	20	20	20	20	20	20	1	1	1	1	1	1	1

Chow parameters $n \leqslant 7$:

For the 247) entries that constitute the Chow parameters for $n \leqslant 7$ reference made to:

WINDER,R.O."THRESHOLD FUNCTIONS THROUGH n=7"

Scientific Report No.7 A.F.C.R.L.

Contract AF 19 (604)-8423.October 1964.

APPENDIX C : CANONIC SPECTRA OF BOOLEAN FUNCTIONS , $n \leqslant 4$,

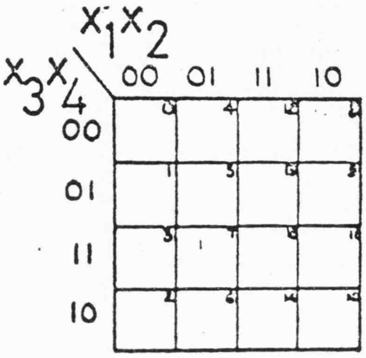
UNDER TRANSLATIONAL-EQUIVALENCE.

Fn. No.	0	1	2	3	4	12	13	14	23	24	34	123	124	134	234	1234	ORDER n	
1	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T
2	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	T
3	14	2	2	2	2	-2	-2	-2	-2	-2	-2	2	2	2	2	-2	4	T
4	2	14	2	2	2	-2	-2	-2	2	2	2	-2	-2	-2	2	-2	4	T
5	0	12	4	4	4	0	0	0	0	0	0	-4	-4	-4	4	0	4	T
6	12	4	4	4	0	-4	-4	0	-4	0	0	4	0	0	0	0	3	T
7	4	12	4	4	0	-4	-4	0	4	0	0	-4	0	0	0	0	3	T
8	10	6	6	2	2	-6	-2	-2	-2	-2	2	2	2	-2	-2	2	4	T
9	6	10	6	2	2	-6	-2	-2	2	2	-2	-2	-2	2	-2	2	4	T
10	2	10	6	6	2	-2	-2	2	2	-2	-2	-6	-2	-2	2	2	4	T
11	0	8	8	8	0	0	0	0	0	0	0	-8	0	0	0	0	3	T
12	8	8	8	0	0	-8	0	0	0	0	0	0	0	0	0	0	2	T
13	8	8	4	4	4	-4	-4	-4	0	0	0	0	0	0	-4	4	4	T
14	4	8	8	4	4	-4	0	0	0	0	-4	-4	-4	0	0	4	4	T
15	0	8	8	4	4	0	4	-4	-4	4	0	-4	-4	0	0	0	4	
16	6	6	6	6	6	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	6	4	T
17	2	6	6	6	6	-6	2	2	2	2	-6	-2	-2	-2	-2	2	4	
18	4	4	4	4	4	4	4	-4	-4	4	-4	-4	-4	4	4	-4	4	
	0	1	2	3	4	12	13	14	23	24	34	123	124	134	234	1234	Spectral Coeff.	

T; THRESHOLD

APPENDIX D: 0 1 SPACES ON THE KARNAUGH MAP OF THE LINEAR BOOLEAN FUNCTIONS

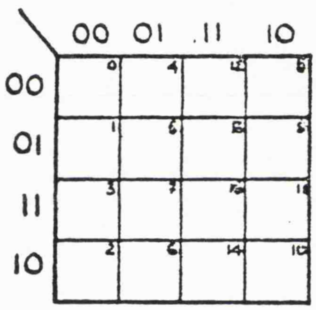
Karnaugh maps of all fourth-order Rademacher/Walsh functions in the range 0,1.



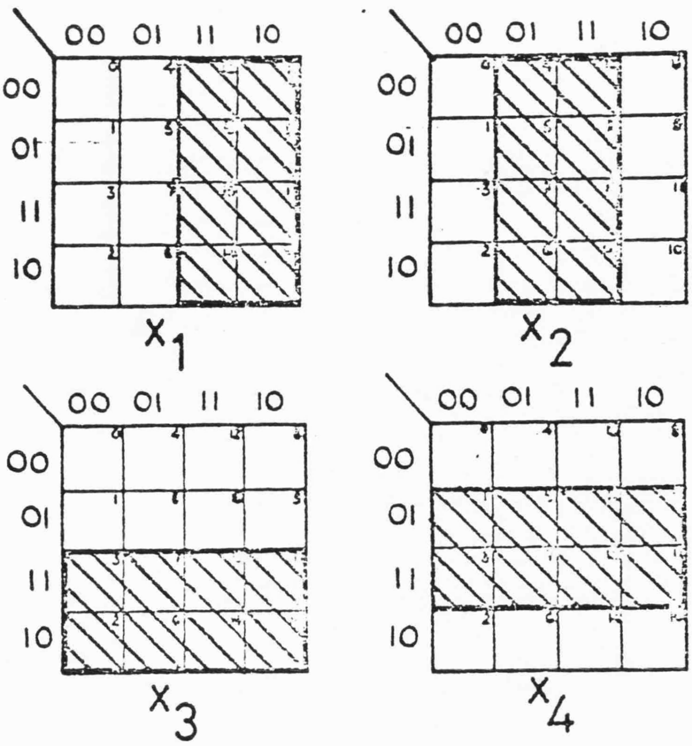
Shaded: TRUE

Blank: FALSE

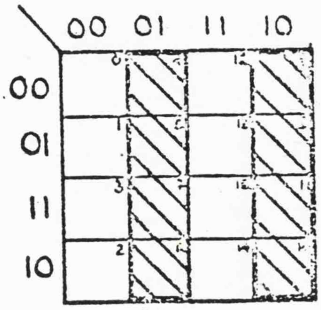
KEY



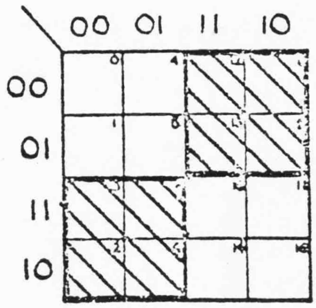
x_0



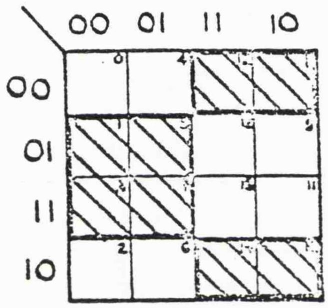
Linear Boolean functions continued,



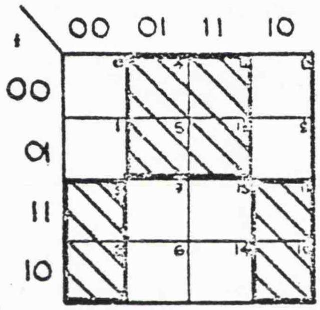
x_{12}



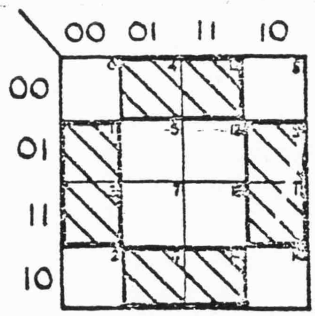
x_{13}



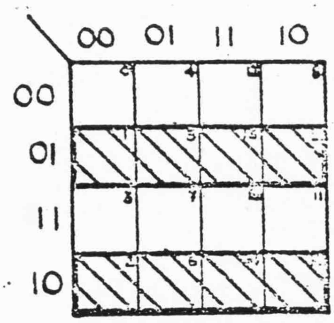
x_{14}



x_{23}

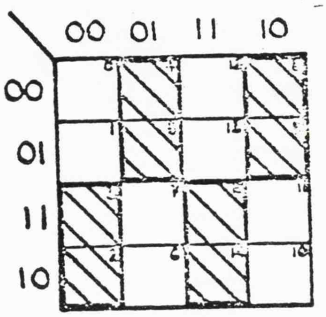


x_{24}

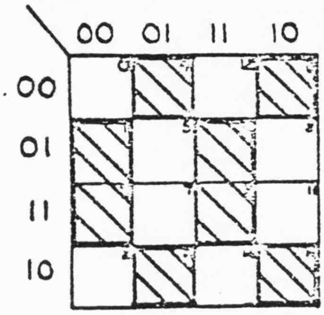


x_{34}

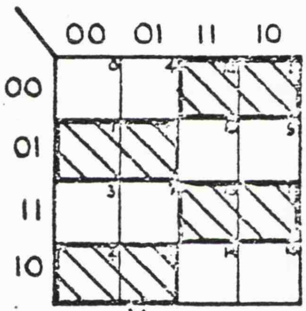
Linear Boolean functions continued,



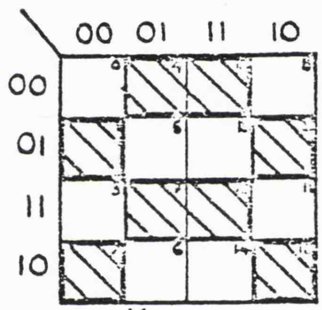
x_{123}



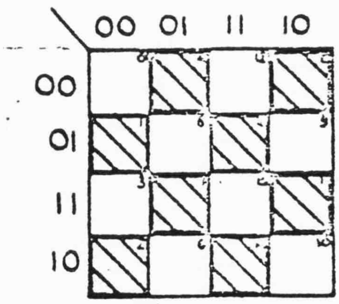
x_{124}



x_{134}



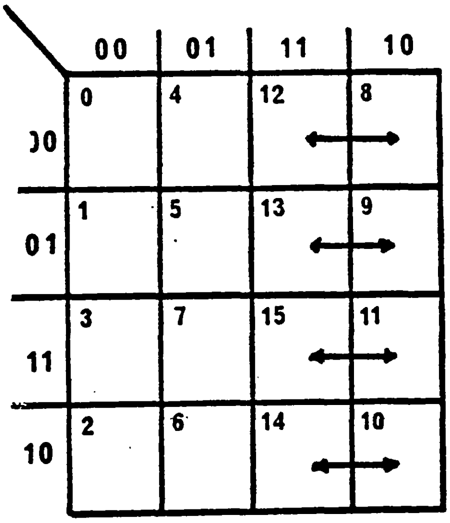
x_{234}



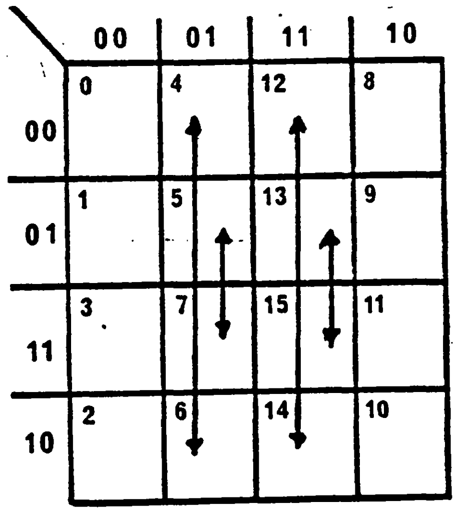
x_{1234}

APPENDIX E : SECOND ORDER SPECTRAL TRANSLATION

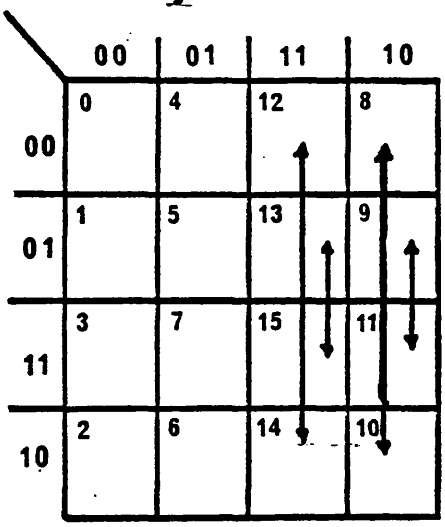
In this Appendix the Karnaugh maps show the minterm-interchanges when the second order spectral coeffecients repaced with the first order ones in the spectrum of a fourth order Boolean function.



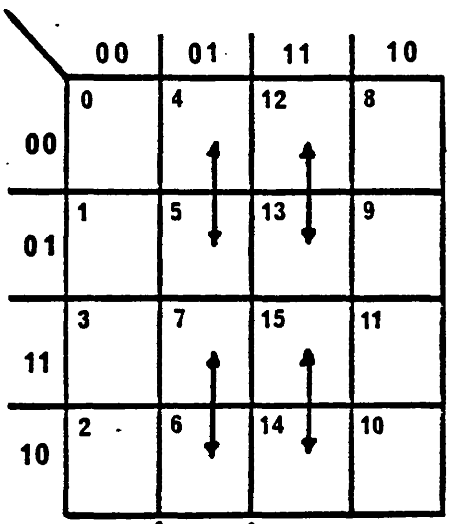
$(1,2) \leftrightarrow 2$



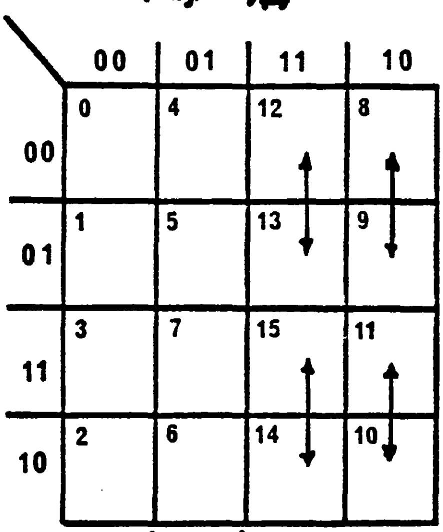
$(2,3) \leftrightarrow 3$



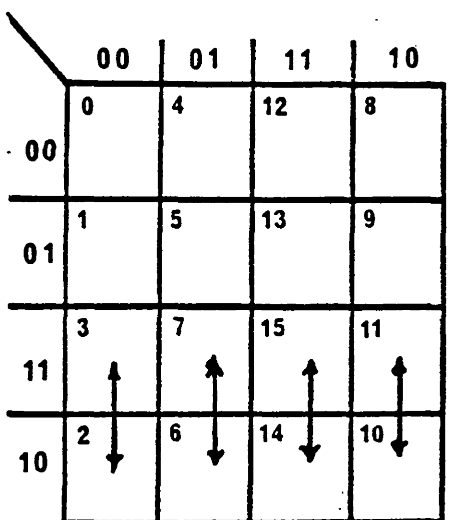
$(1,3) \leftrightarrow 3$



$(2,4) \leftrightarrow 4$

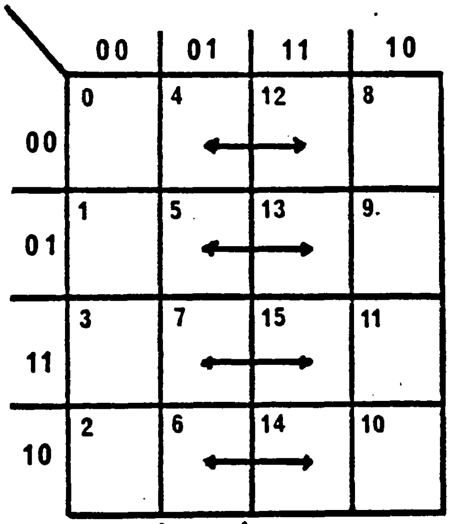


$(1,4) \leftrightarrow 4$

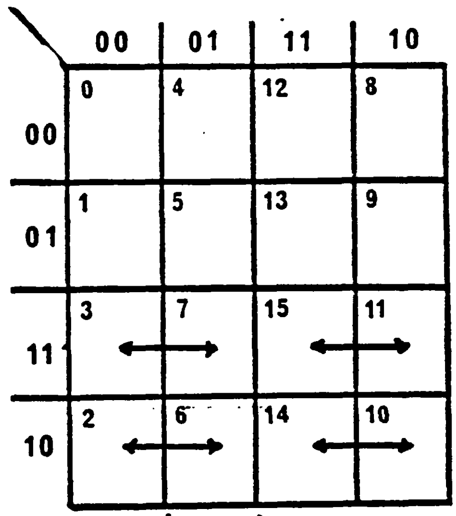


$(3,4) \leftrightarrow 4$

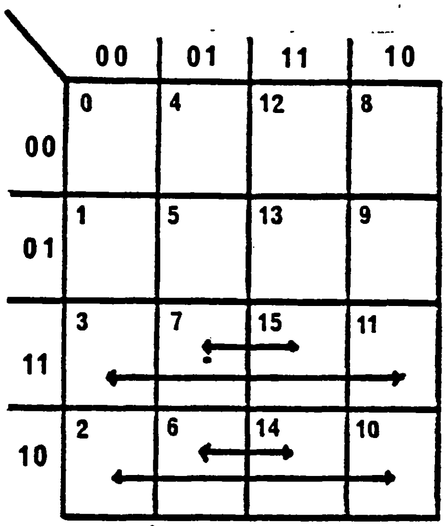
second order spectral translations continued,



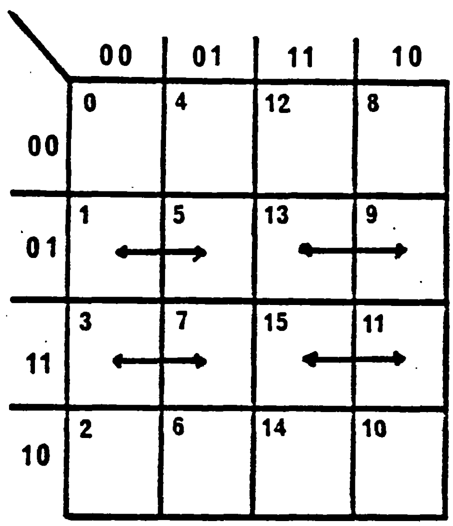
$(1,2)_{\leftrightarrow}1$



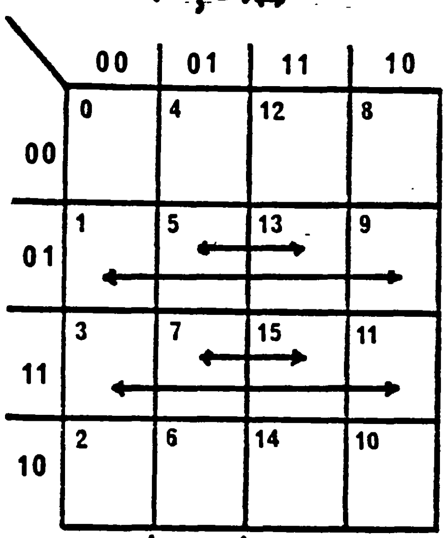
$(2,3)_{\leftrightarrow}2$



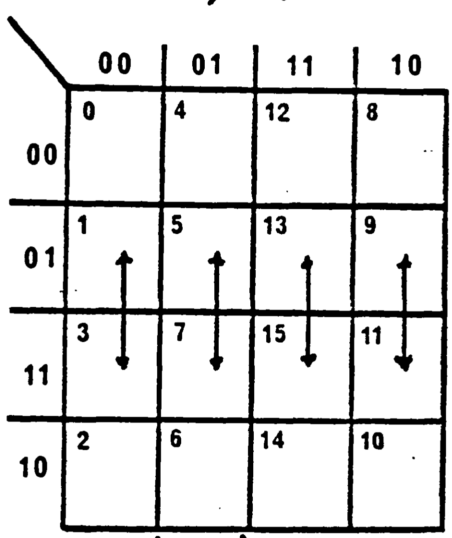
$(1,3)_{\leftrightarrow}1$



$(2,4)_{\leftrightarrow}2$



$(1,4)_{\leftrightarrow}1$



$(3,4)_{\leftrightarrow}3$

APPENDIX F

Paper 1 : Relationship Between Rademacher-Walsh Spectra of Boolean Functions

Reprinted from : Computers and Digital Techniques
May 1978,Vol.1 No.2

Relationships between Rademacher-Walsh spectra of Boolean functions

E. Eris

Indexing terms: *Boolean Functions, Transforms*

Abstract: The relationships between the Rademacher-Walsh spectra of Boolean functions and the spectrum of the Boolean product (AND) and sum (OR) of such functions is investigated. Appropriate matrix operations in the spectral domain are defined for these Boolean operations, and further developments considered.

List of symbols

A_i = ⟨0, 1⟩ vector $\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ of the truth table of a Boolean function A_i, in decimal order

B_i = ⟨1, −1⟩ vector $\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$ of the truth table of a Boolean function A_i, in decimal order

A_i' = Not A_i function

T = Radermacher-Walsh transform of order $n \times n$

S_i = spectrum of the function A_i

[diag. A_i] = diagonal matrix, the elements along the diagonal being those of A_i

$\mathbf{1}$ = unit column vector

I = unit matrix

1 Introduction

The transformation of conventional binary data (normally expressed by truth tables, Boolean equations or state tables involving the two numbers 0 and 1) into some alternative mathematical domain, not confined to two numbers, has received noticeable attention in recent years. The data in such an alternative mathematical domain are normally referred to as the spectrum of the given binary data, and consist of a range of even-integer numbers ranging between -2^n and $+2^n$, where n is the number of independent binary variables in the given data. The transform between these two domains is made by an appropriate orthogonal transform, for example the Rademacher-Walsh transform, which yields the Rademacher-Walsh spectrum of the given binary data. This may be expressed mathematically by

$$TB = S$$

where T = appropriate matrix transform
B = given binary data, appropriately expressed as a single-column vector of two numbers
S = resultant spectrum of the given binary data.

The inverse of the transform enables the binary data to be obtained from given spectral data, i.e.

$$T^{-1}S = B$$

where T^{-1} is the inverse transform of T.

It may be shown that the values of the coefficients in the spectrum represent correlation coefficients showing 'how like' the given binary function is to its inputs and to combinations of its inputs. Further, as the spectrum contains all the information present in the given binary function, but enumerated differently, the spectral coefficient values may themselves be used for logic synthesis purposes,[1-7] and for other applications such as Boolean-function classification[4-6,10] and fault testing of logic networks.[8,11]

As a very simple example to illustrate the basic transform and resulting spectrum, take the two-variable function $f(x) = [x_1 + \bar{x}_2]$. Its output in conventional truth-table order is therefore (1, 0, 1, 1). Converting 0 to +1 and 1 to −1 gives the revised binary-data truth-table $B =$ (−1, 1, −1, −1). Hence the transform of this binary data into the spectral domain using the appropriate Rademacher-Walsh transform is as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ -2 \\ -2 \end{bmatrix}$$

The resulting coefficient values (−2, 2, −2, −2) constitute the spectrum of $f(x)$, and uniquely and fully define it. No other function $f(x)$ can have these particular values and signs. The full meaning of the values may be found in References 1 and 8.

Whilst the mathematical transforms between the binary and spectral domains are rapidly executed by appropriate fast-transform techniques,[2,9] in many practical situations Boolean functions arise which are subsequently connected together by logical operators such as AND or OR. It is, therefore, desirable to be able to execute such logical operations in the spectral domain, using the spectra of the functions being logically connected, rather than having to transform back to the Boolean domain each time such an operation is necessary. This paper therefore discloses the mathematical procedures necessary for combing spectral data to correspond to such logical operations.

2 Boolean product (AND)

The relationship between the product-function spectrum S_p and the individual spectra S_1 and S_2 is given by

$$S_p = T[\text{diag.} A_1]\,T^{-1}S_2 + \tfrac{1}{2}S_1 + \tfrac{1}{2}T\mathbf{1} \qquad (1a)$$

$$= T[\text{diag.} A_2]\,T^{-1}S_1 + \tfrac{1}{2}S_2 + \tfrac{1}{2}T\mathbf{1} \qquad (1b)$$

Paper T115C, first received 14th June and in revised form 30th September 1977

Mr. Eris is with the School of Electrical Engineering, University of Bath, Bath BA2 7AY, England

0140–1335/78/115C–0045 $1·50/0

Proof

The spectra of the product function A_p and the individual functions are given by definition as follows: $S_p \triangleq TB_p$, $S_1 \triangleq TB_1$ and $S_2 \triangleq TB_2$. Using the linear relationships between $\langle 0, 1\rangle$ and $\langle 1, -1\rangle$ domain and denoting AND by $\wedge$,

$$B = -2A + 1 \tag{2a}$$

$$A = -\tfrac{1}{2}B + \tfrac{1}{2}1 \tag{2b}$$

$$A_p \triangleq A_1 \wedge A_2 \equiv [\text{diag.}\, A_1]A_2$$

therefore

$$B_p = -2[\text{diag.}\, A_1](-\tfrac{1}{2}B_2 + \tfrac{1}{2}1) + 1$$

and

$$S_p \triangleq TB_p$$
$$= -2T[\text{diag.}\, A_1](-\tfrac{1}{2}T^{-1}S_2 + \tfrac{1}{2}1) + T1$$

therefore

$$S_p = T[\text{diag.}\, A_1]T^{-1}S_2 - TA_1 + T1$$

But

$$-TA_1 + T1 = -T(-\tfrac{1}{2}T^{-1}S_1 + \tfrac{1}{2}1) + T1$$
$$= \tfrac{1}{2}S_1 + \tfrac{1}{2}T1$$

therefore

$$S_p = T[\text{diag.}\, A_1]T^{-1}S_2 + \tfrac{1}{2}S_1 + \tfrac{1}{2}T1$$

Changing designations in the above will correspondingly yield the proof of eqn. 1*b*.

Example

Given

$$A_1 = \begin{bmatrix}1\\0\\1\\1\\0\\0\\0\\1\end{bmatrix} \text{ and } A_2 = \begin{bmatrix}0\\1\\0\\1\\1\\1\\0\\1\end{bmatrix}$$

then

$$A_p = A_1 \wedge A_2 = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\1\end{bmatrix}$$

Similarly

$$S_1 = \begin{bmatrix}0\\-4\\4\\0\\0\\-4\\-4\\0\end{bmatrix} \text{ and } S_2 = \begin{bmatrix}-2\\2\\-2\\6\\2\\2\\-2\\2\end{bmatrix} \text{ implies } S_p = \begin{bmatrix}4\\0\\4\\4\\0\\0\\-4\\0\end{bmatrix}$$

Hence, confirming S_p using say eqn. 1*b**

$$S_p = T\{\text{diag.}\,[0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1]\}\tfrac{1}{8}T^t S_1 + \tfrac{1}{2} S_2 + \tfrac{1}{2}T1$$

$$T\begin{bmatrix}0\\1\\0\\-1\\1\\1\\0\\-1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}-2\\2\\-2\\6\\2\\2\\-2\\2\end{bmatrix} + \frac{1}{2}\begin{bmatrix}8\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

$$= \begin{bmatrix}1\\-1\\5\\1\\-1\\-1\\-3\\-1\end{bmatrix} + \begin{bmatrix}-1\\1\\-1\\3\\1\\1\\-1\\1\end{bmatrix} + \begin{bmatrix}4\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}4\\0\\4\\4\\0\\0\\-4\\0\end{bmatrix}$$

The commutative and associative properties enable this relation to be extended to more than two functions.

3 Boolean sum (OR)

The relationship between the sum function A_s, spectrum S_s, and the individual functions A_1, A_2, spectra S_1, S_2, is given by

$$S_s = T[\text{diag.}\, A_1']T^{-1}S_2 + \tfrac{1}{2}S_1 - \tfrac{1}{2}T1 \tag{3a}$$

or

$$S_s = T[\text{diag.}\, A_2']T^{-1}S_1 + \tfrac{1}{2}S_2 - \tfrac{1}{2}T1 \tag{3b}$$

* Note, the inverse T^{-1} of the $n \times n$ orthogonal Rademacher-Walsh transform T is given by $\dfrac{1}{n} \times T^t$. Also recall that the 8×8 Rademacher-Walsh transform is

$$\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix}$$

Proof

Denoting OR by $\vee$, since $A_s \triangleq A_1 \vee A_2 = (A_1' \wedge A_2')'$ and for any given function A, $S_A = -S_{A'}$ (Reference 6) we obtain from eqn. 1*a*

$$S_s = -\{T[\text{diag.} A_1']\, T^{-1} S_{A_2'} + \tfrac{1}{2} S_{A_1'} + \tfrac{1}{2} T1\}$$

$$= T[\text{diag.} A_1']\, T^{-1} S_2 + \tfrac{1}{2} S_1 - \tfrac{1}{2} T1$$

Proof of eqn. 3*b* follows similarly.

The commutative and associative properties enable this relation to be extended to more than two functions.

4 Development to include exclusive-OR relations

The relationships between spectral coefficients S include the following:

$$S_p = -\tfrac{1}{2} S_e + \tfrac{1}{2} S_1 + \tfrac{1}{2} S_2 + \tfrac{1}{2} T1 \tag{4a}$$

$$S_s = \tfrac{1}{2} S_e + \tfrac{1}{2} S_1 + \tfrac{1}{2} S_2 - \tfrac{1}{2} T1 \tag{4b}$$

where S_e is the spectrum of the exclusive-OR function $A_1 \oplus A_2$.

Proof

$$S_p \triangleq TB_p$$

$$= T\{-2[\text{diag.} A_1] A_2 + 1\}$$

$$= T\{-2(-\tfrac{1}{2}[\text{diag.} B_1] + \tfrac{1}{2} I)(-\tfrac{1}{2} B_2 + \tfrac{1}{2} 1) + 1\}$$

$$= T\{\tfrac{1}{2}(-[\text{diag.} B_1] B_2 + [\text{diag.} B_1]\, 1 + IB_2 - 1) + 1\}$$

$$= \tfrac{1}{2} T\{-[\text{diag.} B_1] B_2 + B_1 + B_2 - 1\} + T1$$

$$= -\tfrac{1}{2} T[\text{diag.} B_1] B_2 + \tfrac{1}{2} TB_1 + \tfrac{1}{2} TB_2 + \tfrac{1}{2} T1$$

Since it is well known that multiplication of elements of B is equivalent to the exclusive OR of elements of A then

$$S_p = -\tfrac{1}{2} S_e + \tfrac{1}{2} S_1 + \tfrac{1}{2} S_2 + \tfrac{1}{2} T1$$

Further, using $A_1 \vee A_2 = (A_1' \wedge A_2')'$ and $S_A = -S_A'$ and eqn. 4*a*

$$S_s = \tfrac{1}{2} S_e + \tfrac{1}{2} S_1 + \tfrac{1}{2} S_2 - \tfrac{1}{2} T1$$

as

$$A_1' \oplus A_2' = A_1 \oplus A_2.$$

Example

A_1 and A_2 as in the previous example; then

$$A_e \triangleq A_1 \oplus A_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad B_e = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \quad S_e = \begin{bmatrix} -2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ 2 \\ 2 \end{bmatrix}$$

using, say, eqn. 4*a*

$$S_p = -\frac{1}{2}\begin{bmatrix} -2 \\ -2 \\ -6 \\ -2 \\ 2 \\ -2 \\ 2 \\ 2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ -4 \\ 4 \\ 0 \\ 0 \\ -4 \\ -4 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} -2 \\ 2 \\ -2 \\ 6 \\ 2 \\ 2 \\ -2 \\ 2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 4 \\ 4 \\ 0 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

The commutative and associative laws enable this exclusive-OR development to be extended to cater for more than two functions. It can be proved that for three functions the above relationships become

$$S_p = \frac{1}{2^2}\{S_{e_{123}} - (S_{e_{12}} + S_{e_{13}} + S_{e_{23}}) + (S_1 + S_2 + S_3)\} + \tfrac{3}{4} T1 \tag{6a}$$

$$S_s = \frac{1}{2^2}\{S_{e_{123}} + (S_{e_{12}} + S_{e_{13}} + S_{e_{23}}) + (S_1 + S_2 + S_3)\} - \tfrac{3}{4} T1 \tag{6b}$$

where $S_{e_{123}}$ is the spectrum of the function $A_1 \oplus A_2 \oplus A_3$ and so on.

Finally adding eqns. 4*a* and 4*b* we obtain

$$S_p + S_s = S_1 + S_2 \tag{7}$$

5 Conclusions

Mathematical methods of combining spectral data which enable Boolean operations to be performed in the spectral domain, without having to transform back to the two-valued binary domain, have been given. The availability of such methods is increasingly desirable as more information and experience are gained in methods of logic design and logic analysis using spectral rather than binary data. The logical operations of AND, OR etc. are of course basic to the buildup of all logic networks other than extremely trivial situations.

It may possibly be thought that the processes discussed above, which are necessary to manipulate and combine the spectral data in order to perform logical operations, are tedious – however, such manipulations of numerical data prove to be more readily executed by computer aid than the corresponding manipulation of binary data using Boolean relationships. This is particularly true as the size and complexity of the problem increase. Hence it is confidently predicted that the methods discussed in this short paper will find increasing use in c.a.d. situations[1,3,11] where logic-network design is being performed by spectral techniques rather than by conventional binary synthesis methods.

6 Acknowledgments

I wish to thank S.L. Hurst and C.R. Edwards for their helpful discussions. The work reported here is being pursued under a Turkish Government student grant, 1977.

7 References

1 'Recent developments in digital logic design'. Proceedings of conference at School of Electrical Engineering, University of Bath, UK, Sept. 1977

2 LECHNER, R.J.: 'Harmonic analysis of switching functions', *in* MUKHOPADHYAY, A. (Ed.): 'Recent developments in switching theory' (Academic Press, New York, 1971)

3 KARPOVSKY, M.G.: 'Finite orthogonal series in the design of digital devices' (John Wiley, N.Y., 1976)

4 EDWARDS, C.R.: 'The application of the Rademacher-Walsh transform to Boolean function classification and threshold-logic synthesis', *IEEE Trans.*, 1975, C-24, pp. 48–62

5 HURST, S.L.: 'The application of Chow parameters and Rademacher-Walsh matrices in the synthesis of binary functions', *Comput. J.*, 1973, 16, pp. 165–173

6 EDWARDS, C.R.: 'Matrix methods in combinational logic design'. Ph.D. thesis, University of Bath, 1973

7 DERTOUZOS, M.L.: 'Threshold logic: a synthesis approach'. Research Monograph 32, MIT Press, 1965

8 HURST, S.L.: 'Testing logic networks', *Wireless World*, 1977, 83, pp. 82–86

9 WALLIS, J.S.: 'Hadamard matrices', (Springer-Verlag, Lecture notes 292 N.Y., 1972)

10 GOLOMB, S.B.: 'On the classification of Boolean functions', *IRE Trans.*, 1959, CT6, special supplement, pp. 176–186

11 EDWARDS, C.R.: 'The design of easily tested circuits using mapping and spectral techniques', *Radio & Electron. Engin.*, 1977, 47, pp. 321–342

APPENDIX F Continued :

Paper 2 : A Minterm Interchange Operation in the Walsh Spectrum Domain

Reprinted from : Electronics Letters
16 th February 1978. Vol.14 No.4

A MINTERM INTERCHANGE OPERATION IN THE WALSH SPECTRUM DOMAIN

Indexing terms: Boolean functions, Transforms, Walsh functions

A mathematical operation is investigated in the Walsh spectrum domain which corresponds to the interchange of any minterms of a Boolean function.

List of symbols:

$F = (+1, -1)$ column vector $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_\nu \end{bmatrix}$ of the truth table of an n variable binary function f, in decimal order, where $\nu = 2^n - 1$

F' = The function determined by any minterm interchange of the function F

$T = 2^n \times 2^n$ (orthogonal) Rademacher-Walsh transform in Hadamard order

$P = 2^n \times 2^n$ (orthogonal) permutation matrix, defining the interchange of the minterms of a function (in the case considered this matrix is symmetric)

S, S' = Hadamard-ordered spectra of the functions F, F' respectively

Introduction: The transformation of conventional binary data into a spectral domain and the possible use of this spectral data for logic synthesis has recently been investigated, the transform between these two domains being made by an appropriate orthogonal transform such as the Rademacher-Walsh transform or Hadamard transform.[1-5] It is important to investigate the correspondence between operations in the Boolean domain and spectral domain if spectral data is to be used for logic design purposes. Some of these relationships have been investigated previously and applied to logic circuit synthesis.[1,3,6]

In this letter a further spectrum operation which corresponds to the interchange of any minterms of a Boolean function is investigated.

Relationships between the spectra of the functions F and F':

$$F' \triangleq PF \tag{1}$$

$$S \triangleq TF, \quad F = T^{-1}S \tag{2}$$

$$S' \triangleq TF' \tag{3}$$

From eqns. 1-3 we develop the similarity transform between these two spectra

$$S' = TPF$$

where

$$S' = TPT^{-1}S \tag{4}$$

Calculation of the TPT^{-1} similarity transform: Calculation of the similarity transform TPT^{-1} matrix for the interchange of any two minterms proceeds as follows: Since the permutation matrix P is real 2^n-square-symmetric, there exists a real orthogonal matrix A such that[8]

$$A^{-1}PA = [\operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_\nu)]$$

where $(\lambda_0, \lambda_1, \ldots, \lambda_\nu)$ are all eigenvalues of the P matrix, $\nu = 2^n - 1$. The matrix A is determined by the eigenvectors associated with the eigenvalues $(\lambda_0, \lambda_1, \ldots, \lambda_\nu)$ of the permutation matrix P. Defining

$$Z \triangleq TA, \quad Z \triangleq \begin{bmatrix} z_{00}\, z_{01} \ldots z_{0\nu} \\ \vdots \\ z_{\nu_0}\, z_{\nu_1} \ldots z_{\nu\nu} \end{bmatrix} \triangleq [Z_0 Z_1, \ldots, Z_\nu]$$

then

$$T = ZA^{-1}$$

$$T^{-1} = \frac{1}{2^n}T^t = \frac{1}{2^n}(ZA^{-1})^t = \frac{1}{2^n}(ZA^t)^t = \frac{1}{2^n}AZ^t$$

whence

$$TPT^{-1} = \frac{1}{2^n}Z[\operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_\nu)]Z^t \tag{5}$$

Z is an orthogonal square matrix because both T and A matrices are orthogonal. A typical entry t_{kl} of TPT^{-1} would be

$$t_{kl} = \frac{1}{2^n}\sum_{m=0}^{\nu} \lambda_m z_{km} z_{lm} \tag{6}$$

On the one hand there exists a relation,

$$Z_k Z_l^t = [z_{k0}, \ldots, z_{k\nu}][z_{l0}, \ldots, z_{l\nu}]^t = \begin{cases} 0 & \text{if } k \neq l \\ 2^n & \text{if } k = l \end{cases}$$

where both Z_k and Z_l are the columns of Z, because Z is an orthogonal matrix. On the other hand it can easily be shown that the permutation matrix which interchanges two minterms of a Boolean function F, will have all eigenvalues of value $+1$ except one which is equal to -1. For example, consider $\lambda_m = -1$, then

$$(\lambda_0, \lambda_1, \ldots, \lambda_m, \ldots, \lambda_\nu) = (1, 1, \ldots, -1, \ldots, 1)$$

Including the above relations in eqn. 6

$$t_{kl} = \begin{cases} \frac{1}{2^n}\{-2(z_{km} z_{lm})\} & k \neq l \\ \frac{1}{2^n}\{2^n - 2(z_{km})^2\} & k = l \end{cases} \tag{7}$$

$$0 < k < \nu; \quad 0 < l < \nu$$

Hence, we may conclude that in order to calculate the TPT^{-1} matrix, all we need is just the $Z_m = [z_{0m}, z_{1m}, \ldots, z_{\nu m}]^t$ column vector which is determined by $Z_m = TA_m$, where A_m is the eigenvector associated with the eigenvalue $\lambda_m = -1$ of the P matrix. Then, with U = unit matrix

$$TPT^{-1} = \frac{1}{2^n}\{2^n U - 2Z_m Z_m^t\}$$

If the permutation matrix P interchanges ith and jth components of F, we will find the A_m eigenvector[8] associated with $\lambda_m = -1$

$$[P - \lambda U]\big|_{\lambda=-1} A_m = 0$$

$$A_m = [a_0, a_1, \ldots, a_q, \ldots, a_\nu]^t$$

$$a_q = \begin{cases} 0 & \text{if } q \neq i \quad q \neq j \\ \frac{1}{\sqrt{2}}\left(\text{or} -\frac{1}{\sqrt{2}}\right) & \text{if } q = i \\ -\frac{1}{\sqrt{2}}\left(\text{or} \frac{1}{\sqrt{2}}\right) & \text{if } q = j \end{cases} \tag{8}$$

Then Z_m column vector can be determined

$$Z_m = TA_m = \mp \frac{1}{\sqrt{2}} [R_{i_m} - R_{j_m}] \tag{9}$$

where R_{i_m} and R_{j_m} are both Rademacher-Walsh functions which correspond in position to the ith and jth components (or minterms) of the function F. Finally we obtain

$$TPT^{-1} = U - \frac{1}{2^n}\{[R_{i_m} - R_{j_m}][R_{i_m} - R_{j_m}]^t\} \tag{10}$$

Considering the effect on the kth component s_k of the spectrum of the function F, since for a typical entry h_{uv} in the position u and v in the Hadamard ordered matrix T

$$h_{uv} = (-1)^{\sum_{i=0}^{\nu} u_i v_i} \tag{11}$$

where u_i and v_i are the ith bit in the binary representation of integers u and v respectively,[9] we develop from the above eqn. 10

$$\begin{aligned} s_k' &= s_k - \frac{1}{2^n}\{r_{ki_m} - r_{kj_m}\}[R_{i_m} - R_{j_m}]^t S \\ &= s_k - \frac{1}{2^n}\{((-1)^{I_m^t K} - (-1)^{J_m^t K}) \; [R_{i_m} - R_{j_m}]^t\} S \end{aligned} \tag{12}$$

where I_m, J_m and K column vectors are binary representations of the integers i_m, j_m, and k. i_m and j_m show the interchanged minterms numbers; k represents the considered spectrum component number.

Generalisation of any two-minterms interchange to any number of minterms interchange: For the above situation the permutation matrix has all +1 on its diagonal except on the two rows corresponding to the interchanged minterms. Those two rows had off-diagonal +1's which were symmetric to the diagonal. Consider the pairwise interchanging of p minterms, each pair-space disjoint, p any even number $2 < p < \frac{1}{2}2^n$. Now the permutation matrix would have +1's on its diagonal except the p number of rows corresponding to the interchanged minterms. These rows would have off-diagonal symmetric +1 values two-by-two. Since the P matrix is orthogonal[8] characteristic roots have absolute value +1. Using the known mathematical relationship[10]

$$\frac{a_\nu}{a_{\nu+1}} = (-1) \sum_{l=0}^{\nu} \lambda_l \tag{13}$$

between the roots and the coefficients of a polynomial such as $a_{\nu+1}\lambda^{\nu+1} + a_\nu \lambda^\nu + \ldots + a_0$, and the relationship

$$\frac{a_\nu}{a_{\nu+1}} = (-1) \sum_{l=0}^{\nu} P_{ll} \tag{14}$$

between the characteristic equation coefficients and the related matrix terms,[7] where P_{ll} is a diagonal term, we obtain

$$\sum_{l=0}^{\nu} \lambda_l = \sum_{l=0}^{\nu} P_{ll} \tag{15}$$

Suppose ν = total number of eigenvalues, c = number of eigenvalues of value -1, d = number of eigenvalues of value $+1$, from the last eqn. 15

$$\sum_{l=0}^{\nu} \lambda_l = d - c = \nu - p$$

since $\nu = c + d$, we find $c = p/2$, namely

$$(\lambda_0, \lambda_1, \ldots, \lambda_c, \ldots, \lambda_\nu) = (\underbrace{-1, -1, \ldots, -1}_{p/2}, 1, 1, \ldots, 1)$$

The typical term of TPT^{-1} becomes

$$t_{kl} = \begin{cases} \frac{1}{2^n}\left\{-2 \sum_{m=1}^{c} z_{km} z_{lm}\right\} & k \neq l \\ \frac{1}{2^n}\left\{2^n - 2 \sum_{m=1}^{c} z_{km}^2\right\} & k = l \end{cases} \tag{16}$$

$$0 < k < \nu, \quad 0 < l < \nu$$

and

$$TPT^{-1} = U - \frac{2}{2^n}\{Z_0 Z_0^t + \ldots + Z_m Z_m^t + \ldots + Z_c Z_c^t\} \tag{17}$$

It is known from the eqn. 9

$$Z_m = TA_m = \mp \frac{1}{\sqrt{2}} [R_{i_m} - R_{j_m}], \quad m = 1, 2, \ldots, c$$

where A_m is an eigenvector associated with the eigenvalues of value -1. Finally we obtain

$$TPT^{-1} = U - \frac{1}{2^n} \sum_{m=1}^{c} [R_{i_m} - R_{j_m}][R_{i_m} - R_{j_m}]^t \tag{18}$$

where R_{i_m} and R_{j_m} are Rademacher-Walsh functions which correspond in position to the interchanged mth pair-space disjoint minterms pair.

Considering the effect on the kth component of the spectrum of the function F, using eqn. 11, we can obtain

$$s_k' = s_k - \frac{1}{2^n}\left\{ \sum_{m=1}^{c} \{(-1)^{I_m^t K} - (-1)^{J_m^t K}\} \times [R_{i_m} - R_{j_m}]^t \right\} S \tag{19}$$

Conclusions: The operation described in this paper, namely the interchange of any pair of minterms, and, by extension, more than one pair of minterms, may be employed in synthesis procedures for combinatorial logic networks, where the synthesis is pursued in the spectral domain.[3,4] It is further envisaged that this interchange operation may be employed to facilitate the manipulation of sequential machine state tables.

Acknowledgments: I wish to thank S. L. Hurst and C. R. Edwards for their helpful suggestions and discussions. The work reported here is being pursued under a Turkish Government Grant 1977.

E. ERIS
School of Electrical Engineering
University of Bath
Bath BA2 7AY, England

22nd December 1977

References

1 LECHNER, R. J.. 'Harmonic analysis of switching functions', *in* MUKHOPADHYAY, A. (Ed.). 'Recent developments in switching theory' (Academic Press, 1971)
2 KARPOVSKY, M. G.: 'Finite orthogonal series in the design of digital devices' (Wiley. 1976)
3 EDWARDS, C. R.: 'The application of Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis', *IEEE Trans.*, 1975, C-24, pp. 48–62
4 Proceedings of the conference on Recent developments in digital logic design. School of Electrical Engineering, University of Bath, UK, September 1977
5 WALLIS, J. S.: 'Hadamard Matrices'. Lecture notes 292 (Springer-Verlag, 1972)
6 EDWARDS, C. R.: 'Matrix methods in combinational logic design'. Ph.D Thesis, University of Bath, 1973
7 HOHN, F E.: 'Elementary matrix algebra' (MacMillan, 1964), p. 280
8 AYRES, F. J. R.: 'Matrices' (Schaum, 1962)
9 AHMED, N., SCHREIBER, H. H., and LOPRESTI, P. V.: 'On the notation and definition of terms related to a class of complete orthogonal functions', *IEEE Trans.*, 1973, EMC-15, pp. 75–80
10 KORN, G. A., and KORN, T. M.: 'Mathematical Handbook for Scientists and Engineers' (McGraw-Hill, 1968 2nd edn.), p. 16

APPENDIX F Continued:

Joint paper : State Assignment and Entropy

Reprinted from : Electronics Letters
22 nd June 1978 Vol.14 No.13

STATE ASSIGNMENT AND ENTROPY

Indexing terms: State assignment. Sequential circuits

A recent paper has presented a method of state assignment based upon the maximisation of true (or false) minterms in the next-state functions. This method has the advantage of very elegant implementation. This letter seeks to interpret these results in the light of the concept of entropy. It is shown that there is some justification for using the maximisation of true (false) minterms for optimal state assignment and some indication is given of the limitations of the method.

Introduction: A recent letter by Lala[1] has proposed a method of state assignment where the number of true (or false) minterms in the next-state functions is maximised. It is stated that this method gives good results in practice but no theoretical justification is presented.

In essence the method seeks to associate the maximum number of true (false) minterms with the most frequently occurring states in the state table. This has the effect of producing next-state functions with a predominance of true (false) minterms. The method, it is claimed,[1] gives solutions which, costwise, agree substantially with other, more complex, assignment procedures.

This is an intriguing result since it does not seem intuitively obvious that functions with very small or very large ratios of true/false minterms will necessarily be easier to synthesise than those having small ratios (except in extreme cases).

One method of rationalising this problem is to introduce the concept of function entropy.

Function entropy: The concept of function entropy[2,3] is important in digital synthesis because there is a close relationship between the entropy of a function and the cost of implementing the function in terms of logic hardware.[4-6]

The characteristics of the average cost (the diode count) of the two-level minimal form of a single-output combinatorial logic network which implements a Boolean function f of n variables and u 1s have been investigated by Cook and Flynn.[5] The entropy of f is defined as

$$H(f) = \frac{u}{2^n}\log_2\frac{2^n}{u} + \frac{(2^n-u)}{2^n}\log_2\frac{2^n}{(2^n-u)} \equiv H(n,u) \qquad (1)$$

where the minimum (average) cost of implementing such a function can be expressed as[5]

$$C(n,u)_{av} = K(n)\,H(n,u) \qquad (2)$$

Fig. 1 shows the actual diode cost[7] of randomly chosen combinatorial Boolean functions; from Kellerman[3] and Cook and Flynn[5] for $n \leq 6$. Each point represents the average cost of 20 minimised circuits.

The $C(6,u)_{av}$ curve for $K(6) = 74{\cdot}6$* is also shown. The very close agreement between the practical results and the function given by eqn. 2 should be noted. Fig. 1 also shows the upper cost limit which, for a 2-level diode implementation, is generated when all true (false) minterms are disjoint. The total cost is then given by

$$C_{max} = \begin{cases} u(n+1) & 0 < u \leq 2^{(n-1)} \\ (2^n-u)(n+1) & 2^{(n-1)} < u \leq 2^n \end{cases} \qquad (3)$$

since n diodes are required for each of u AND gates and one u-input OR gate is also required. The lower cost limit (C_{min}) for $n \leq 6$ is also shown.

It is important to note that, from the results of Kellerman,[3] the general form of eqn. 2 is also applicable to minimised circuits fabricated with vertex and exclusive-OR gates. In addition the diode cost is closely related to the cost of implementing memories of the f.p.l.a. type.

* See Mase[8] for a discussion of scaling factors

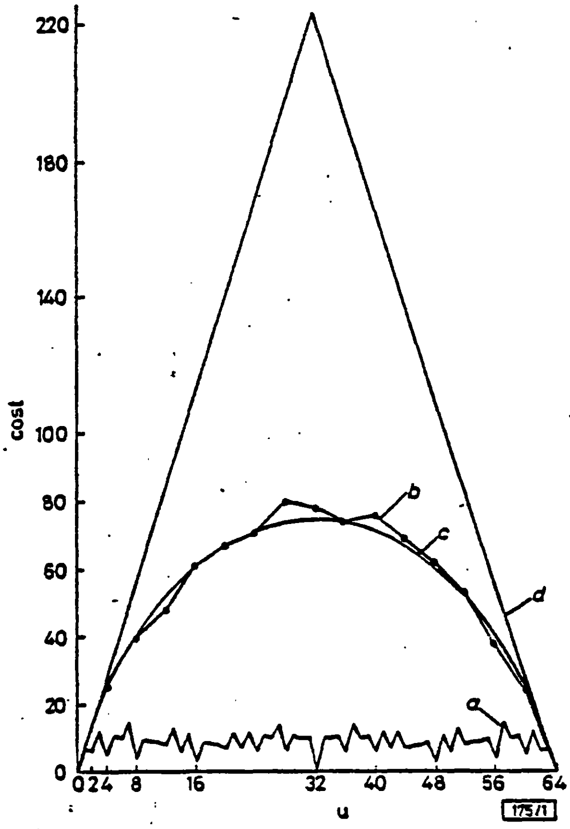


Fig. 1 *Cost against the number of 1s (u) on an n = 6 order map. (Diode ccts.)*

a C_{min}
b C_{av} practical results from Kellerman[3]
c $K(6)\,H(n,u)$ with $K(6) = 74{\cdot}6$ (Scaled entropy)
d C_{max}

Lala's method and entropy: The method of Lala,[1] which seeks to maximise the ratio of true/false minterms in the next state functions can now be interpreted in terms of the above results.

Clearly, by maximising the number of true (false) minterms in each function the *average* cost of circuit fabrication (given by the entropy function) is reduced. In addition the maximum cost also becomes smaller. It is important to remember, however, that this is a *statistical* result. That is, given a large number of sequential machines, *chosen at random*, the average cost of implementing the next-state functions is reduced by maximising the numbers of true (false) minterms thereof.

Lala[1] has stated that this method has been applied to state tables of differing complexity and gives results comparable to and in some cases better than those obtained by applying some of the (more complex) published methods.

In using Lala's method we feel that the following observations may prove instructive:

(*a*) In practice sequential machines are, by virtue of their purpose, highly structured. This leads to the surmise that the factor $K(n)$ in eqn. 2 is generally smaller for a sequential machine next-state function than would be the case for a randomly generated Boolean function.

(*b*) The method is likely to produce good results only at reasonably high minterm ratios (say above 80%). With small ratios the method may generate particularly costly results.

(*c*) To quantify the above statistic an investigation into the distribution of cost about C_{av} for each u is required.

Acknowledgments: The authors are respectively supported under UK SRC and Turkish Government grants.

C. R. EDWARDS
E. ERIS

School of Electrical Engineering
University of Bath
Claverton Down
Bath
England

25th May 1978

References

1 LALA, P. K.: 'An algorithm for the state assignment of synchronous sequential circuits', *Electron. Lett.*, 1978, 14, pp. 199–201
2 SHANNON, C. E.: 'A mathematical theory of communication', *Bell Syst. Tech. J.*, 1948, 27, pp. 379–423, 623–656
3 KELLERMAN, E.: 'A formula for logical network cost'. *IEEE Trans.*, 1968, C-17, pp. 881–884
4 KELLERMAN, L.: 'A measure of computational work'. *ibid.*, 1972, C-21, pp. 439–446
5 COOK, R. W., and FLYNN, M. J.: 'Logic network cost and entropy', *ibid.*, 1973, C-22, pp. 823–826
6 MASE, K.: 'Comments on 'A measure of computational work' and 'Logic network cost and entropy' ', *ibid.*, 1978, C-27, pp. 94–95
7 PHISTER, M. Jun.: 'Logical design of digital computers' (Wiley, New York, 1959), p. 62

0013-5194/78/1175-0390 $1.50/0